Confidence-Based Power-Efficient Framework for Sleep Stage Classification on Consumer Wearables

Hsu-Chao Lai, Po-Hsiang Fang, Yi-Ting Wu, Lo Pang-Yun Ting, and Kun-Ta Chuang
Department of Computer Science and Information Engineering
National Cheng Kung University, Tainan, Taiwan
Email: {hclai, phfang, ytwu, lpyting}@netdb.csie.ncku.edu.tw
ktchuang@mail.ncku.edu.tw

***Abstract*—Consumer wearable devices like smartwatches enable real-time tracking of vital signs with various sensors. Accordingly, experts may leverage smart home techniques for treating sleep disorders by targeting specific sleep stages. To facilitate this scenario, this paper focuses on sleep stage classification based on body movement and heart rate signals detected by wearables in real time. Due to the limited battery capacity, it is crucial to balance the trade-off between power efficiency and classification accuracy. To address the problem, inspired by multi-tasking, we propose COPS, an innovative framework that includes a power-efficient shallow classifier for simple cases and a deep classifier for complex instances. COPS introduces an intelligent switch, CESwitch, to determine a confidence score that directs input to either the shallow or deep classifier. By selectively activating the shallow classifier, the overall expected power consumption could be lower. Two strategies of CESwitch, namely Confidence Delegation and Agreement Verification, are proposed and examined. Notably, both COPS and CESwitch can be seamlessly integrated with existing deep sleep stage classifiers. Comprehensive experimental results on two real datasets manifest that COPS outperforms state-of-the-art lightweight and deep sleep stage classifiers by reducing 77.8% computational cost in terms of FLOPs with only 1.9% accuracy drop. Moreover, adapting to an existing deep model saves up to 32.7% in FLOPs compared to its original architecture.**

***Index Terms*—Sleep Stage Classification, Power Efficiency, Confidence-based Strategy.**

I. Introduction

Consumer wearable devices like Google Fitbit, Apple Watch, and Garmin smartwatches have revolutionized personal healthcare by enabling continuous, non-invasive monitoring of physiological parameters in real time. These parameters, including heart rate, blood oxygen levels, skin temperature, and movement, empower various healthcare applications. These applications help users easily manage and monitor a range of health conditions, such as sleep quality, cardiac syndromes, and mobility issues, by integrating advanced machine learning models. According to a survey of 9,303 US adults supported by the National Institutes of Health [1], over 29% Americans used wearable devices, with 82% of them using these devices for health purposes in 2020. By April 2022, the population had increased to 45%, with 92% of users reporting that they used these devices to maintain and manage their health [2].

With regard to sleep, which significantly influences memory, immune function, cognitive processes, and emotional regulation, it is crucial to monitor sleep quality for maintaining physiological and psychological well-being [3]–[5]. However, conventional tools may fall short of providing consistent monitoring in real time. For example, patients may have to fill out self-rated questionnaires to obtain the Pittsburgh Sleep Quality Index [6] or visit medical centers for electroencephalography (EEG), electrocardiography (ECG), and electrooculography (EOG) reports, which require specialized and expensive medical equipment.

In contrast, consumer wearables such as smartwatches and smart wristbands offer affordability, convenience, and continuous monitoring [7]. These devices leverage AI models to identify sleep stages using accessible sensors like accelerometers (ACC) and photoplethysmography (PPG), enabling the monitoring of sleep patterns in natural environments rather than clinical settings [8] [9]. As a result, the detected sleep patterns and sleep stages are more accurate, identifying sleep disorders precisely. Combining this accuracy with real-time monitoring, these consumer devices show great potential in promptly and consistently controlling other AIoT and smart home devices [10] [11] to apply advanced sleep modulation techniques, such as optogenetics [12] and acoustic stimulation [13], by targeting specific sleep stages.

To facilitate this scenario, given three-dimensional accelerators and heart rate signals collected by wearable devices in real time, *our goal is to accurately classify each period of sleep patterns into different sleep stages while maintaining low computational costs*, owing to limited battery capacity. The sleep stages are labeled by Wake, Rapid Eye Movement (REM), N1, N2, and N3 according to the American Academy of Sleep Medicine (AASM) standards [14]. This problem raises two main issues that motivate our model design:

(i) Trade-off between accuracy and power consumption: To achieve high classification accuracy, state-of-the-art models employed deep structures [15]–[17] to fully extract hidden information from signals. However, they required extensive computation, typically measured in Floating Point Operations per Second (FLOPs) [18] [19], which could significantly drain the limited battery capacity of wearable devices and potentially shorten their lifespan. Conversely, reducing model complexity without careful design or directly using lightweight models [18] [19] may lead to relatively inaccurate predictions. Therefore, it is crucial to balance the trade-off between accuracy and power consumption.

(ii) Generality across different classifiers: Many existing sleep stage classifiers are power-hungry and are not designed for the limited battery life of consumer wearables. Furthermore, the architectures and feature extraction techniques differ significantly among various manufacturers, complicating the implementation of a universal solution for optimizing battery efficiency. Therefore, a general power-efficient framework that can be integrated into existing models without requiring extensive modification is desirable. The generality is crucial for advancing the field and enabling more widespread and practical use of sleep stage classifiers in consumer wearables.

To address the aforementioned challenges, this paper proposes an innovative multi-task framework, *Confidence-based Power-efficient Sleep Stage Classifier* (*COPS*), for classifying sleep stages using a hierarchical structure. COPS incorporates a shared feature extraction architecture and further exploits two towers: *a deep classifier for handling complex input instances and a shallow classifier for simpler ones*.

For challenge (i), we introduce a novel module, namely *Confidence-based Energy-saving Switch* (*CESwitch*), to dynamically determine which classifier to activate for each instance. By appropriately activating the shallow classifier, the approach lowers the total expected FLOPs, resulting in energy efficiency. Two different strategies of CESwitch, *Confidence Delegation* and *Agreement Verification*, are introduced to address different scenarios. Note that both the multi-tasking framework and CESwitch are adaptive and compatible with a wide range of existing models, effectively addressing the challenge (ii). This adaptability ensures that the framework can be integrated into various architectures without requiring significant modifications, thus enhancing its practicality.

Comprehensive experimental results on two real datasets manifest that COPS outperforms state-of-the-art lightweight and deep sleep stage classifiers by reducing 77.8% computational cost in terms of FLOPs with only 1.9% accuracy drop. Moreover, adapting COPS to an existing deep model saves up to 32.7% in FLOPs compared to its original architecture. Our contributions can be summarized as follows:

- A new power-efficient framework, COPS, is proposed to reduce power consumption in sleep stage classification by incorporating both shallow and deep classifiers. It enables a dynamic balance between energy savings and accuracy.
- An innovative confidence-based switching mechanism, CESwitch, intelligently navigates input instances to appropriate classifiers. Two strategies, CD and AV, are introduced for different conditions.
- Comprehensive experimental results on two real datasets demonstrate that COPS significantly reduces computational costs in FLOPs while maintaining similar accuracy compared to baselines.
- Experimental results also validate that adapting COPS to existing deep sleep stage classifiers requires little modification and effectively saves power, highlighting the generality.

The rest of this paper is organized as follows. Section II reviews state-of-the-art literature and justifies the necessity of this paper. Section III presents details of COPS and CESwitch. Section IV further demonstrates their generality and adaptability. Section V presents the experimental results and discusses the performance. Finally, Section VI concludes this paper.

II. Related Work

A. Deep Sleep Stage Classifiers

While data from accelerometers (ACC) and photoplethysmography (PPG) sensors are relatively simple for the classification of sleep stages compared to professional medical equipment, deep learning has emerged as a powerful tool for extracting informative knowledge and identifying patterns in consumer wearables. AccSleepNet [15] leveraged self-attention blocks to analyze cross-axis attention from three-dimensional movement data. Olsen et al. [16] employed U-Net, a deep auto-encoder-like structure, to extract spatio-temporal features from ACC and PSG signals at different resolutions. Fonseca et al. [17] used multiple residual blocks to extract features and further classified sleep stages with stacked bidirectional GRU layers. Overall, these deep models empowered wearable devices to perform sleep stage classification with easily accessible data. However, the complexity of these models makes them less suitable for deployment on wearable devices due to their high power consumption.

B. Lightweight Sleep Stage Classifiers

Professional devices that measure EEG, ECG, and EOG provide direct insights into sleep stages, showing better potential of using power-efficient models. DeepSleepNet [20] used two towers of convolution layers to learn latent features and stacked bidirectional LSTMs to classify sleep stages, resulting in approximately 21M parameters. CNN-Transformer [21] reduced this to around 380K parameters by eliminating one tower and replacing the bidirectional LSTMs with a transformer unit. LightSleepNet [18] employed channel shuffle and Global Average Pooling, further reducing parameters to 43.8K. Micro SleepNet [19] incorporated Efficient Channel Attention to a similar structure, enhancing accuracy while slightly increasing parameters to 48.2K. However, these methods all require professional equipment for input, limiting their application in cost-effective commercial wearable devices. On the other hand, despite their advances in reducing model size, they still consistently consume power even when users are clearly awake, which is potentially detectable by using simpler and more power-efficient methods. These issues justify the need for this paper to further reduce unnecessary power consumption.

III. Confidence-based Sleep Stage Classification

Given three-dimensional ACC and PSG signals, the time series data are truncated by every 30 seconds. This is a standard preprocess [18] [19] [21] to better identify local features for real-time inference according to AASM [14]. The proposed COPS aims to classify sleep stages of each segment while reducing the computational load of deep models by intelligently activating low-cost shallow classifiers when

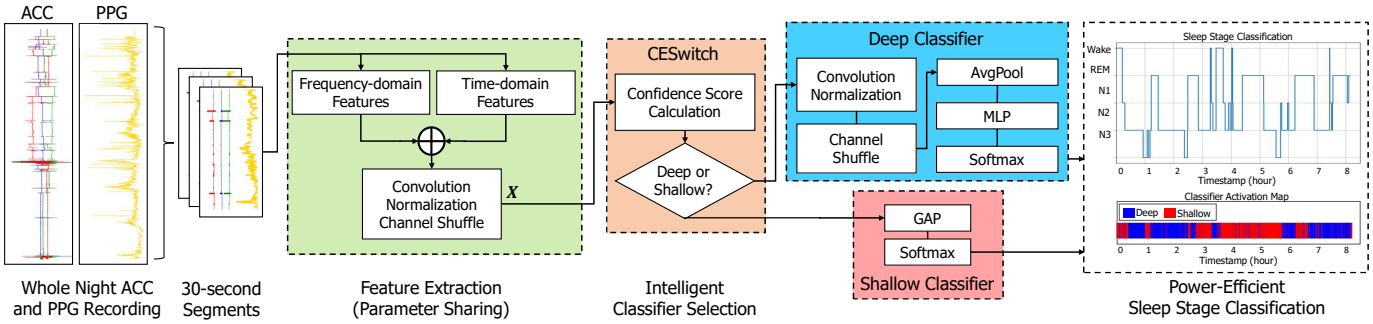


Fig. 1. Illustration of Confidence-based Power-efficient Sleep Stage Classifier (COPS).

appropriate, as illustrated in Figure 1. Specifically, the idea is to assign simple cases to the shallow classifier (red area), while more challenging cases are handled by the more computationally intensive deep classifier (blue area). In addition, inspired by multi-task learning, COPS implements parameter sharing between the shallow and deep classifiers (green area), avoiding redundant process and further enhancing efficiency.

To maintain classification accuracy with smaller and power-efficient models, COPS exploits a novel confidence-based controller module, Confidence-based Energy-saving Switch (CESwitch), deciding which classifier to activate (orange component). If CESwitch is confident in the shallow classifier, it outputs its prediction directly, thus minimizing additional computational overhead. Otherwise, the deep classifier is engaged. In the following, the structures of the shallow and deep classifiers are introduced in Section III-A. Section III-B further details the two strategies of CESwitch. Finally, the process of feature extraction is presented in Section III-C.

A. Shallow and Deep Classifiers

Note that it is not necessary to activate deep classifiers all the time. For instance, while awake, people generate obvious patterns in ACC and PPG signals that can be easily identified. Consistently running deep classifiers in such cases would result in unnecessary energy consumption. Instead, COPS proposes to employ an additional shallow classifier, which requires significantly less computational power and is sufficient to handle these easy tasks.

The red area in Figure 1 illustrates the shallow classifier component in COPS. It conducts a global average pooling (GAP) layer [18] [19] followed by a softmax function in sequence to predict the class of the input signal:

$$\begin{aligned} \mathbf{X}_{\text{gap}} &= GAP(\mathbf{X}) \\ \mathbf{P}_{\text{shallow}} &= softmax(\mathbf{W}_{\text{shallow}}\mathbf{X}_{\text{gap}} + \mathbf{b}_{\text{shallow}}), \end{aligned} \tag{1}$$

where $\mathbf{X}$ denotes the time-domain and frequency-domain features extracted and polished at the green area (details in Section III-C). $\mathbf{X}_{\text{gap}}$ is the normalized features. $\mathbf{P}_{\text{shallow}}$ is the vector containing the softmax probability distribution of each class based on a linear transformation, with $\mathbf{W}_{\text{shallow}}$ and $\mathbf{b}_{\text{shallow}}$ representing the weight matrix and bias vector, respectively. The final output, denoted by y^*_{shallow}, is the class having the highest softmax probability:

$$y^*_{\text{shallow}}, p^*_{\text{shallow}} = \arg\max_c \mathbf{P}_{\text{shallow}},$$

where c is the index of entry of the softmax distribution. p^*_{shallow} denotes the greatest softmax probability among all c, which is also the probability of y^*_{shallow}.

On the other hand, to handle difficult cases that the shallow classifier cannot easily address, the deep classifier (blue area) incorporates additional group convolution layers for more in-depth feature analysis. Similar to the shallow classifier, a softmax layer is appended at the end to predict the class, and the output class with the greatest likelihood is denoted by y^*_{deep}.

For the objective function, the classes of sleep stages include Wake, REM, N1, N2, and N3 in our datasets. It is worth noting that the number of each class is not equal, facing the imbalance issue. As a result, COPS uses the Class-Balanced Softmax cross-entropy loss $\mathcal{L}$ [22] as follows:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\frac{1-\beta}{1-\beta^{n_c}} y_{i,c}\log(\hat{y}_{i,c}),$$

where C and N denote the number of classes and samples, respectively. In this context, $y_{i,c}$ indicates whether class c is assigned to the i-th sample, and $\hat{y}_{i,c}$ is the predicted probability for class c. n_c represents the number of samples of class c in the current batch, and $\beta \in [0, 1)$ is a hyperparameter that reweights the loss value based on the inverse frequency of class occurrence. We further duplicate $\mathcal{L}$ to $\mathcal{L}_{\text{shallow}}$ and $\mathcal{L}_{\text{deep}}$ with respect to the shallow and deep classifiers. The overall loss, denoted by $\mathcal{L}_{\text{total}}$, is the sum of $\mathcal{L}_{\text{shallow}}$ and $\mathcal{L}_{\text{deep}}$.

Equipped with both the shallow and deep classifiers, COPS offers the flexibility and adaptability in choosing appropriate classifiers based on the task at hand. The shallow classifier efficiently saves power while handling simpler tasks, whereas the deep classifier accurately addresses more complex input cases. However, choosing the right classifier is not trivial. Incorrectly activating the shallow classifier can compromise accuracy, while falsely using the deep classifier can lead to unnecessary power consumption. In the next section, we introduce our strategy for making the right choice.

B. Confidence-based Energy-saving Switch (CESwitch)

To effectively balance accuracy and computational costs as depicted by challenge (i), COPS employs CESwitch to derive confidence scores in using the shallow classifier rather than the deep one. In this section, we introduce two different strategies for calculating the confidence scores: *Confidence Delegation* and *Agreement Verification*, respectively.

1) Confidence Delegation (CD): Since the last layer already computes the likelihood of each class with the softmax function in Eq. (1), a straightforward confidence score for the shallow classifier can be set to the likelihood of the output class: p^*_{shallow}. The greater p^*_{shallow} is, the more confidence we are in the output class y^*_{shallow}. However, some input signals may be too ambiguous for the shallow classifier to discriminate effectively due to its simple structure, causing a smaller p^*_{shallow}. In such cases, delegating the task to the deep classifier is expected to be more appropriate.

Accordingly, CESwitch using Confidence Delegation strategy is designed to be an if-else gate:

$$\text{CESwitch-CD}(p^*_{\text{shallow}}) = \begin{cases} \text{shallow} & \text{if } p^*_{\text{shallow}} \geq \theta \\ \text{deep} & \text{otherwise} \end{cases}, \quad (2)$$

where θ is a predefined confidence threshold. In other words, if the likelihood exceeds the confidence threshold θ, y^*_{shallow} is considered convincing enough to be outputted, thus saving energy. Lowering θ may increase power-saving efficiency, as the shallow classifier would be used more frequently, but it could potentially compromise accuracy. Detailed comparisons are provided in the experimental sections.

2) Agreement Verification (AV): While CD needs experienced experts to set θ, AV strategy introduces a data-driven approach from a different aspect: it activates the shallow classifier only when the deep classifier is likely to agree with it on prediction results, i.e., $y^*_{\text{shallow}} = y^*_{\text{deep}}$. Otherwise, the deep classifier makes the predictions.

To learn the likelihood of agreement between the deep and shallow classifiers for the input feature $\mathbf{X}$, Agreement Verification strategy is designed to be a binary classifier as follows:

$$\text{CESwitch-AV}(\mathbf{X}) = y_{\text{AV}} = LR(MLP(\mathbf{X})), \quad (3)$$

where LR and MLP are the logistic regression layer and the multi-layer perceptron layer, respectively. y_{AV} denotes the confidence score, verifying their agreement based on $\mathbf{X}$. To train the logistic regression, the ground truth $y^{\text{truth}}_{\text{AV}}$ are collected by:

$$y^{\text{truth}}_{\text{AV}} = \begin{cases} \text{shallow} & \text{if } y^*_{\text{shallow}} = y^*_{\text{deep}} \\ \text{deep} & \text{otherwise} \end{cases}. \quad (4)$$

As a result, AV does not validate predicted sleep stages directly. Instead, AV avoids unnecessary computational costs on the deep classifier when the shallow one is likely to produce the same prediction results in a data-driven way.

The training process of this binary agreement classifier is not concurrent with the training of the multi-tasking classifiers. It is trained separately, after the multi-tasking classifiers have completed their training, to collect ground truth by assessing their agreement on the training data. Therefore, using the AV strategy does not increase the difficulty of training the multi-tasking classifiers.

3) Comparing CD and AV: CD and AV strategies have distinct advantages and disadvantages in calculating confidence scores. CD is cost-effective to implement but requires the computation of both classifiers when the confidence score is below θ. Let Ω_{feature}, Ω_{shallow}, and Ω_{deep} denote the required computation with respect to feature extraction, the shallow classifier, and the deep classifier. The expected overall computation of inference for each instance, termed as Ω_{CD}, is:

$$\Omega_{\text{CD}} = \Omega_{\text{feature}} + \Omega_{\text{shallow}} + P_{\text{deep}} \cdot \Omega_{\text{deep}}, \quad (5)$$

where $P_{\text{deep}} = P(p^*_{\text{shallow}} < \theta)$ is the probability of the deep classifier stepping in the prediction task. Since CD uses p^*_{shallow} as the confidence score, Ω_{shallow} must be computed for every input. Minimizing power consumption requires keeping Ω_{shallow} low, but causes the trade-off that a simpler shallow classifier may make incorrect predictions with high confidence.

In contrast, AV is more conservative in classification performance by using the shallow classifier only when the agreement is likely to be verified, which potentially guarantees performance. In addition, while CD derives its confidence scores by solving a five-class problem, AV simplifies this to a binary classification problem. This simplification may reduce the difficulty of delivering a robust confidence score. The expected overall computation of inference for each instance Ω_{AV} is:

$$\Omega_{\text{AV}} = \Omega_{\text{feature}} + \Omega_{\text{bi}} + P_{\text{agree}} \cdot \Omega_{\text{shallow}} + (1 - P_{\text{agree}}) \cdot \Omega_{\text{deep}},$$

where Ω_{bi} is the computational overhead of training an extra binary classifier. Although AV incurs this overhead, its simple structure keeps the extra overhead minimal. P_{agree} represents the likelihood of the two classifiers reaching a consensus. Note that AV activates either the shallow or the deep classifier for any instance, unlike CD with the potential to activate both.

In general, CD strategy is favorable when the shallow classifier is reliable and the difference between Ω_{deep} and Ω_{shallow} is significant. Conversely, the AV approach is favored when achieving the highest performance is crucial, and the device can afford the extra storage needed for the additional binary classifier. Nevertheless, both strategies of CESwitch enable COPS to intelligently reduce power consumption while maintaining good accuracy. Developers may choose their preferred strategies by evaluating the expected computation and accuracy on the validation dataset.

C. Feature Extraction

For each input signal segment, the feature extraction process begins by extracting time-domain and frequency-domain features with grouped convolutions and a Short-Time Fourier Transform (STFT) layer, respectively. Following this, channel shuffling [18] is applied to improve accuracy and reduce computation simultaneously. In addition to the motion and

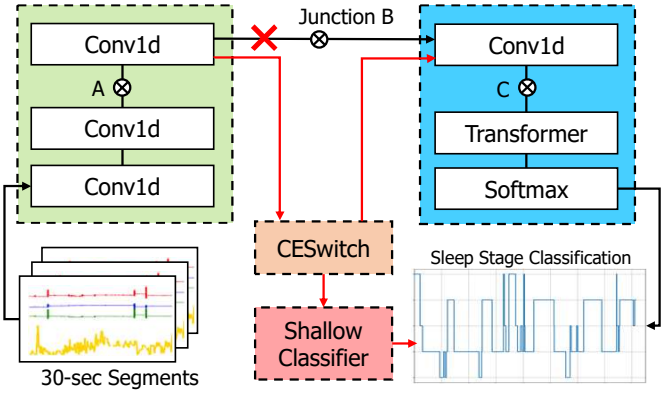


Fig. 2. An illustration of integrating our idea into an existing deep model.

heart rate data, we include the daily timestamp as a feature, capturing temporal behavior in real-world time. The extracted features are shared by the following classifiers, thereby eliminating additional computational costs.

To analyze ACC and PPG data (denoted by $\mathbf{X}_{\text{input}}$), STFT layer has shown its credibility in biomedical non-stationary signal analysis [16] [23] [24]. Unlike traditional Fourier Transform, which gives a global frequency representation, STFT allows for time-frequency localization. Therefore, how the frequencies in our signal change over time is observable. Each channel undergoes the STFT process:

$$\text{STFT}(x[n]) = \sum_{m=0}^{M-1} x[n+m] \cdot w[m] \cdot e^{-j2\pi km/N},$$

where $x[n]$, k, m, and N are the input signal, frequency bin, sample index, and the total number of FFT points, respectively. $w[m]$ denotes the Hann window function [24]. The real part of the STFT output is then extracted to represent the frequency-domain features of user movements and heart rates.

On the other hand, to extract time-domain features $\mathbf{X}_{\text{conv1d}}$, one-dimensional convolution, termed as $Conv1d$, is implemented to unveil relationships within the three-axis acceleration and heart rate data $\mathbf{X}_{\text{input}}$:

$$\mathbf{X}_{\text{conv1d}} = GELU(BN(Conv1d(\mathbf{X}_{\text{input}}))),$$

where BN denotes the batch normalization [18]. $GELU$ is the Gaussian Error Linear Unit activation that weights inputs by their probability under a Gaussian distribution [25], which has shown better performance than ReLU in handling sequential data.

After concatenating the time-domain and frequency-domain features into $\mathbf{X}_{\text{concat}}$, we employ the channel shuffle technique [18] to enhance feature representation by rearranging grouped channels. Let C and G represent the number of channels and groups, respectively. The channels are first divided equally into G groups, each containing C/G channels. The channels within these groups are then shuffled in a structured manner, ensuring that subsequent layers receive a diverse mix of features across the groups. This process allows the model to learn from a diverse set of feature combinations while keeping computational costs low, effectively reducing overfitting and conserving energy.

IV. Adapting to Other Existing Classifiers

We demonstrate the generality of the multi-tasking framework along with CESwitch, as described in challenge (ii), by integrating them into an existing deep sleep stage classifier. Although many manufacturers keep their models confidential, this section can serve as a guideline for incorporating these novel mechanisms into arbitrary deep classifiers.

Figure 2 illustrates the adaption process. The black arrows illustrate the original data flow of the Deep CNN-Transformer model [21], abbreviated as DeepCT hereafter, for sleep stage classification. DeepCT concatenates four convolution layers in sequence to extract features from input signal segments. Following that, a transformer unit is added to explore temporal information. Finally, a softmax layer classifies the sleep stages. DeepCT is a single deep classifier rather than a multi-tasking framework. As a result, even when users are awake and generate obvious movement signals, the deep model continues to run, wasting unnecessary power.

To improve power efficiency, we modify the data flow by removing the original black arrow at junction B and introducing new data flows in red arrows, along with CESwitch and a shallow classifier. Therefore, the new structure matches COPS in Figure 1. The preceding $Conv1d$ layers (green area) and the following layers (blue area) of junction B are now treated as the feature extraction and the deep classifier components in COPS, respectively. CESwitch gets to determine which classifier is activating for each instance, thereby avoiding energy waste.

While existing deep sleep stage classifiers [16] [17] have already identified their junctions between feature extraction layers and deep classification structures, it is possible to append our CESwitch and a shallow classifier at those junctions to achieve our goal. Otherwise, developers may select junctions according to the trade-off: junctions closer to raw data (e.g., junction A) may yield lower feature quality but offer greater energy savings. Conversely, junctions farther from the raw data (e.g., junction C) involve deeper feature extraction layers, which can reduce energy savings but improve feature quality. We demonstrate the effectiveness and the impact of junction selection of adapting DeepCT in Section V-D1.

V. Experimental Results

A. Experimental Setting

1) Dataset: Two public datasets are used to evaluate COPS, CESwitch, and their generality. SleepAccel [26] [27] collects ACC and PPG signals from Apple Watch. Data were collected at the University of Michigan from June 2017 to March 2019, and there are 31 subjects in total. DREAMT [28] is another dataset using an Empatica E4 wristband to collect ACC and PPG data, which are verified by domain experts. A total of 100 unique participants were recruited from the

TABLE I
STATISTICS OF SLEEPACCEL AND DREAMT DATASETS.

Dataset	**Wake**	**REM**	**N1**	**N2**	**N3**
Counts					
SleepAccel	2,133	5,427	1,624	12,184	3,189
DREAMT	19,971	8,331	8,819	39,687	2,661
Percentage (%)					
SleepAccel	8.7%	22.1%	6.6%	49.6%	13%
DREAMT	25.1%	10.5%	11.1%	49.9%	3.3%

TABLE II
BASELINE COMPARISONS ON SLEEPACCEL DATASET.

Method	Accuracy	F1	FLOPs (M)	Parameters (K)
LightSleepNet	0.789	0.705	9.448	64.84
DeepCT	0.802	0.731	9.857	384.84
DeepCT-CD-0.8	0.787	0.710	9.047	385.48
DeepCT-AV	0.778	0.702	8.835	592.46
COPS-CD-0.8	0.727	0.647	2.010	32.01
COPS-AV	0.726	0.642	1.941	238.99

TABLE III
PERFORMANCE OF COPS ON SLEEPACCEL DATASET.

Method	Accuracy	F1	FLOPs (M)	P_{shallow}
COPS-DeepOnly	0.729	0.646	2.033	-
COPS-CD-1	0.728	0.650	2.033	0.02%
COPS-CD-0.9	0.723	0.645	2.026	1.60%
COPS-CD-0.8	0.727	0.648	2.010	4.87%
COPS-CD-0.7	0.716	0.633	1.967	13.04%
COPS-CD-0.6	0.712	0.627	1.898	28.47%
COPS-AV	0.726	0.642	1.941	62.20%

TABLE IV
PERFORMANCE OF COPS ON DREAMT DATASET.

Method	Accuracy	F1	FLOPs (M)	P_{shallow}
COPS-DeepOnly	0.627	0.528	2.033	-
COPS-CD-1	0.620	0.519	2.033	0.0%
COPS-CD-0.9	0.623	0.519	2.026	1.45%
COPS-CD-0.8	0.620	0.517	2.016	3.65%
COPS-CD-0.7	0.622	0.525	2.003	6.21%
COPS-CD-0.6	0.620	0.519	1.981	10.82%
COPS-AV	0.629	0.526	1.943	61.65%

Duke University Health System Sleep Disorder Lab. The total counts and distribution of the data are shown in Table I.

For preprocessing, the frequencies of signals are upsampled to 64Hz using backward fill, by following the procedure in [16]. Interquartile Range (IQR) normalization is applied to the value of the 3-axis acceleration data and heart rate data to mitigate the impact of outliers.

2) Evaluation Metrics: To comprehensively evaluate the performance of baselines, we employ multiple evaluation metrics that capture different aspects of model effectiveness and efficiency. For classification performance, accuracy and macro F1-score are included. Accuracy measures the proportion of correctly classified instances among the total instances. The macro F1-score, defined as the harmonic mean of precision and recall, is first computed separately for each class and then averaged. Compared to micro F1, which assigns equal weight to each data point, the class-level macro F1 is better suited for evaluating imbalanced sleep stage data. We use F1 to represent the macro F1-score hereafter.

On the other hand, to evaluate the power consumption, Floating Point Operations (FLOPs) and the number of model parameters are used. FLOPs measure the computational complexity of the model by counting the number of floating-point operations required to make a prediction. Following [18] [19], it is used to evaluate the computational efficiency and feasibility of deploying the model on resource-constrained devices, e.g., wearable devices in this paper. The number of model parameters is an important metric for understanding the model's complexity and potential overfitting. A model with fewer parameters is preferred in our scenario.

3) Baselines: To examine the generality and effectiveness, we compare with multiple variants of COPS and state-of-the-art sleep stage classifiers:

- *LightSleepNet* [18]: A state-of-the-art lightweight model minimizing the number of parameters, which increases to 64.84K due to the adaption to our input.
- *DeepCT* [21]: The vanilla CNN-transformer in a sequence of four convolution layers and a transformer unit. The parameter number is 384.84K.
- *DeepCT-CD-θ*: DeepCT using CESwitch with Confidence Delegation strategy and different θ. They all have 385.48K parameters.
- *DeepCT-AV*: DeepCT using CESwitch with Agreement Verification strategy. The parameter numbers are 592K and 1007K for single-junction and three-junction settings (detailed later), respectively.
- *COPS-DeepOnly*: COPS excluding the shallow classifier and CESwitch. The parameter number is 31.65K.
- *COPS-CD-θ*: COPS using CESwitch with Confidence Delegation strategy, where different confidence threshold θ will be evaluated. They all have 32.01K parameters.
- *COPS-AV*: COPS using CESwitch with Agreement Verification strategy. The parameter number increases to 238.99K owing to the extra agreement classifier.

Figure 2 illustrates how DeepCT incorporates the shallow classifier along with CESwitch. For DeepCT, the default junction appending CESwitch and the shallow classifier is B since it achieves a balanced result (detailed later). Otherwise, it will be specified along with experimental results.

We employ the Adam optimizer to train every baseline. The learning rate and batch size are 0.01 and 512, respectively. To ensure robust model validation and to prevent overfitting, we use stratified five-fold cross-validation. β in the Class-Balanced Softmax Cross Entropy is 0.99.

In the following, we overview a macro-level comparison of COPS against lightweight and deep models in Section V-B. Detailed comparisons among variants of COPS are then discussed in Section V-C. Section V-D presents the generality of adapting our framework to DeepCT and comparisons among multiple variants. Finally, Section V-E analyzes the behavior of shallow and deep classifiers across different sleep stages.

TABLE V
PERFORMANCE OF DEEPCT ADAPTION ON SLEEPACCEL DATASET.

Junction	Method	Accuracy	F1	FLOPs (M)	P_{shallowA}	P_{shallowB}	P_{shallowC}
A	DeepCT	0.796	0.726	9.857	-	-	-
	DeepCT-CD-1	0.797 (+0.14%)	0.728 (+0.25%)	9.857 (-0.00%)	0.0%	-	-
	DeepCT-CD-0.9	0.792 (-0.55%)	0.721 (-0.73%)	9.635 (-2.07%)	4.06%	-	-
	DeepCT-CD-0.8	0.785 (-1.43%)	0.713 (-0.66%)	9.296 (-5.70%)	10.29%	-	-
	DeepCT-CD-0.7	0.780 (-2.03%)	0.707 (-2.64%)	8.644 (-12.31%)	22.23%	-	-
	DeepCT-CD-0.6	0.767 (-3.67%)	0.692 (-4.72%)	7.845 (-20.41%)	36.87%	-	-
	DeepCT-AV	0.773 (-2.89%)	0.694 (-4.41%)	6.815 (-30.86%)	59.54%	-	-
B	DeepCT	0.802	0.731	9.857	-	-	-
	DeepCT-CD-1	0.795 (-0.82%)	0.726 (-0.67%)	9.857 (-0.01%)	-	0.02%	-
	DeepCT-CD-0.9	0.789 (-1.55%)	0.715 (-2.20%)	9.336 (-5.29%)	-	29.61%	-
	DeepCT-CD-0.8	0.787 (-1.86%)	0.711 (-2.83%)	9.047 (-8.21%)	-	45.98%	-
	DeepCT-CD-0.7	0.780 (-2.72%)	0.703 (-3.82%)	8.831 (-10.41%)	-	58.24%	-
	DeepCT-CD-0.6	0.766 (-4.49%)	0.686 (-6.17%)	8.641 (-12.34%)	-	69.04%	-
	DeepCT-AV	0.778 (-2.99%)	0.703 (-3.82%)	8.835 (-10.37%)	-	69.74%	-
C	DeepCT	0.798	0.722	9.857	-	-	-
	DeepCT-CD-1	0.798 (-0.00%)	0.727 (+0.62%)	9.857 (-1.05%)	-	-	0.06%
	DeepCT-CD-0.9	0.789 (-1.14%)	0.717 (-0.75%)	9.704 (-1.56%)	-	-	30.66%
	DeepCT-CD-0.8	0.787 (-1.40%)	0.710 (-1.70%)	9.622 (-2.39%)	-	-	47.05%
	DeepCT-CD-0.7	0.782 (-1.99%)	0.708 (-1.90%)	9.565 (-2.97%)	-	-	58.47%
	DeepCT-CD-0.6	0.759 (-4.90%)	0.681 (-5.71%)	9.513 (-3.49%)	-	-	68.76%
	DeepCT-AV	0.793 (-0.62%)	0.720 (-0.28%)	9.613 (-2.48%)	-	-	90.13%
A,B,C	DeepCT	0.810	0.738	9.858	-	-	-
	DeepCT-CD-1	0.807 (-0.47%)	0.733 (-0.66%)	9.851 (-0.07%)	0.0%	0.03%	1.28%
	DeepCT-CD-0.9	0.791 (-2.42%)	0.712 (-3.46%)	9.179 (-6.89%)	1.92%	22.56%	35.3%
	DeepCT-CD-0.8	0.769 (-5.13%)	0.693 (-6.14%)	8.828 (-10.49%)	4.83%	34.08%	33.29%
	DeepCT-CD-0.7	0.756 (-6.68%)	0.673 (-8.73%)	8.318 (-15.62%)	12.76%	39.58%	29.34%
	DeepCT-CD-0.6	0.708 (-12.66%)	0.617 (-16.41%)	7.696 (-21.94%)	24.60%	40.13%	22.58%
	DeepCT-AV	0.768 (-5.18%)	0.682 (-7.59%)	6.632 (-32.74%)	57.47%	19.07%	19.31%

B. Baseline Performance Overview

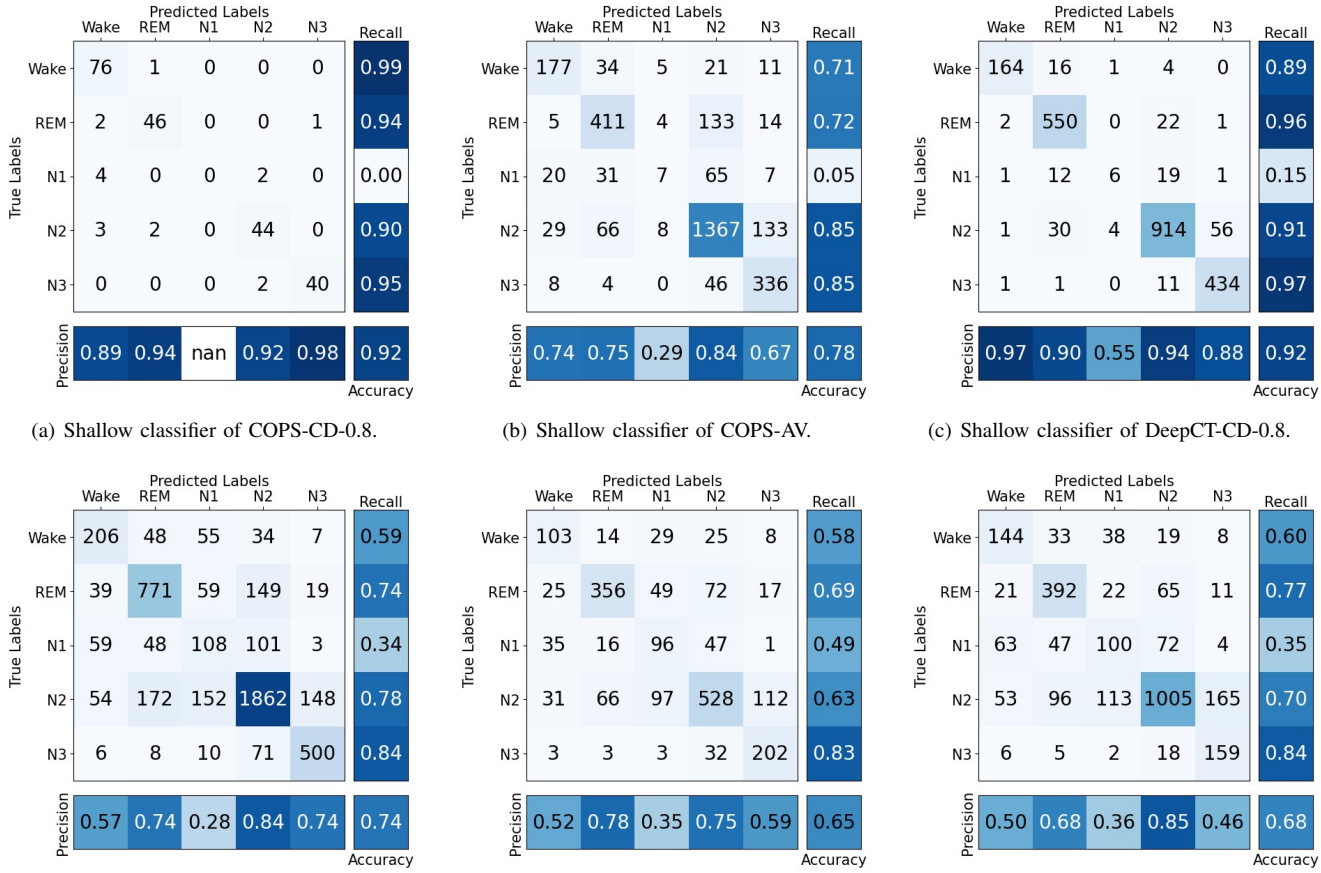
Table II compares lightweight models, deep models, and COPS with different confidence-based power-saving mechanisms on the SleepAccel dataset. COPS-CD-0.8 has the fewest parameters while achieving classification performance comparable to LightSleepNet. Moreover, COPS-AV not only performs similarly but also results in the lowest FLOPs among all. These results manifest the power efficiency of the proposed COPS for the sleep stage classification on consumer wearable devices. In addition, integrating CESwitch into the most sophisticated model, DeepCT, only slightly deteriorates accuracy by 1.9% and 3.0% with respect to CD and AV strategies. Nevertheless, the integration empowers DeepCT to maintain similar classification performance and even achieve lower FLOPs compared to LightSleepNet, highlighting the generality and effectiveness of the proposed framework and mechanisms.

C. Detailed Performance of COPS

Tables III and IV compare the performance of each variant of COPS on the SleepAccel and DREAMT datasets, respectively. COPS-DeepOnly achieves the highest accuracy in both datasets due to its additional layers that deeply recognize hidden features. However, this comes with the cost of increased FLOPs, indicating the highest power consumption. On the other hand, as the *confidence threshold* θ decreases from 1 to 0.6, the ratio of input instances handled by the power-saving shallow classifier P_{shallow} increases from 0.02% to 28.47% on SleepAccel dataset. This increment results in a 6.7% reduction in FLOPs with only a slight decrease in accuracy (2.3%) and F1 score (2.8%). These results manifest that the *Confidence Delegation strategy of CESwitch* effectively balances the trade-off between classification performance and power consumption by avoiding activation of the deep classifier when the shallow classifier's output is sufficiently reliable.

The overall performance of COPS is lower on DREAMT compared to SleepAccel. The major difference is a 32.78-49.33% relative drop in N3 precision. Since N2 and N3 share similar patterns of low body movement frequency and low heart rates, distinguishing between them becomes more challenging. Plus, DREAMT has a significantly lower N3 data ratio, further worsening the learning conditions. The difficulty also affects the effectiveness of the shallow classifier. As θ decreases from 1 to 0.6, the acceptance ratio of the shallow classifier P_{shallow} only rises from 0.0% to 10.82%, indicating that the shallow classifier struggles with accurately classifying sleep stages, leading most inputs to be passed to the deep classifier. Consequently, the reduction in FLOPs is limited to just 2.5%, with only a minor drop in accuracy (1.1%) and F1 score (1.6%). Comparing the results of both datasets suggests that CESwitch-CD is more appropriate when the sleep stages are easier to classify.

On the other hand, using the AV strategy allows COPS-AV to maintain similar performance, ranging from -0.4% to +0.3% in terms of accuracy and F1 scores, while achieving a 4.5% drop of FLOPs due to the significant 61.62-62.20% activation rate of the shallow classifier in both datasets. The high activation rate demonstrates that the binary agreement



(a) Shallow classifier of COPS-CD-0.8. (b) Shallow classifier of COPS-AV. (c) Shallow classifier of DeepCT-CD-0.8.

(d) Deep classifier of COPS-CD-0.8. (e) Deep classifier of COPS-AV. (f) Deep classifier of DeepCT-CD-0.8.

Fig. 3. Comparisons of confusion matrices on SleepAccel dataset.

classifier effectively identifies instances where both classifiers are likely to make the same prediction, as further validated by the 85.6% and 86.2% accuracy of the binary agreement classifier on SleepAccel and DREAMT datasets, respectively. The improvement of FLOPs is not proportional due to the extra binary agreement classification process and the relatively small margin between Ω_{shallow} and Ω_{deep}. However, with the high activation rate, the AV strategy shows great potential in power savings while adapting to much deeper sleep stage classifiers.

D. Detailed Performance of DeepCT Adaption

Table V presents the results of adapting the proposed framework to an existing deep sleep stage classifier DeepCT on SleepAccel dataset. The first column lists the junctions where CESwitch and the shallow classifier are integrated, with junctions A, B, and C illustrated in Figure 2. The last part of the table explores the possibility of adding three individual shallow classifiers on all junctions. In the following, we first examine the impact of applying a single CESwitch at various junctions, and then compare these results with the multi-switch configuration. The distributions of a shallow classifier is used at each junction are denoted by P_{shallowA}, P_{shallowB}, and P_{shallowC}. The numbers in parentheses present the relative reduction percentage compared to the corresponding DeepCT.

1) Single-Switch Setting: The upper part of Table V presents the results using one single shallow classifier. Generally speaking, the CD strategies with various θ and junctions are capable of maintaining classification performance, as evidenced by the reduction of at most -4.90% and -6.17% in terms of accuracy and F1, respectively. For power consumption, although the increase in P_{shallowA} as θ decreases is slower compared to P_{shallowB} and P_{shallowC}, junction A achieves a more rapid reduction in FLOPs. This is because the deeper feature extraction layers at junctions B and C cause higher computational costs (Ω_{feature}), which consumes the overall benefits of using the shallow classifier.

For higher confidence thresholds of 0.8 and 0.9, the shallow classifier at junction B is accepted significantly more often than at junction A, with P_{shallowB} being at least 3.47 times greater than P_{shallowA}, leading to greater reductions in FLOPs. Conversely, junction C yields similar performance and acceptance probabilities to junction B, indicating that pushing the shallow classifier to deeper junctions provides only marginal benefits. Therefore, junction B is a sweet spot to be the final choice for a high confidence threshold. Conclusively, the overall results manifest that integrating our framework with existing deep models is robust and power-saving.

On the other hand, the AV strategy significantly improves the utilization rates of the shallow classifiers, ranging from 59.54% to 90.13%. The classification performance drops at most 2.99% in accuracy and 4.41% in F1 score, which are both marginal. Compared to CD, AV is more efficient in transferring input cases to shallow classifiers while achieving solid classification performance.

2) Multi-Switch Setting: The last part of Table V presents the results of using three individual shallow classifiers on each junction, abbreviated as the multi-switch setting hereafter. Following the data flow in Figure 2, if the shallow classifier at one junction is not accepted, the input progresses to the next junction. If none of the shallow classifiers is accepted, the deep classifier takes over.

From the perspective of power consumption, the multi-switch setting of either CD or AV outperforms the corresponding single-switch setting under an arbitrary confidence threshold. This improvement is due to the multi-switch setting significantly reducing the initiation of the deep classifier. For example, at $\theta = 0.9$, the probability of using the deep classifier in the multi-switch setting is 40.22%, compared to 95.94%, 70.39%, and 69.34% for the single junctions A, B, and C, respectively. When the confidence threshold is relaxed to 0.6, the multi-switch setting activates the deep classifier only 12.69% of the time, causing a 21.94% reduction in FLOPs.

For the AV strategy, the probability of using the deep classifier even further declines to 4.15%. Unlike CD, most of the instances (54.47%) are transferred to the shallow classifier at the earliest junction A, resulting in the most substantial power saving with a 32.74% reduction in FLOPs. Moreover, the agreement rates, which is the accuracy of the binary classifier, are 0.86, 0.87, and 0.90 with respect to junctions A, B, and C, leading to reliable overall sleep stage accuracy with only a 5.18% drop. These results manifest AV could be more effective in multi-switch setting than CD in terms of power efficiency and reliability.

Despite the notable improvement in power consumption, the frequent use of shallow classifiers can increase the risk of misclassification. Consequently, the multi-switch setting experiences a slight deterioration in accuracy and F1 scores compared to any single-switch setting. In summary, this multi-switch setting should be particularly suitable when the vanilla model is extremely deep, which requires an aggressive power-saving mechanism but less concern about potential decline in classification performance.

E. Classifier Behavior Across Sleep Stages

Figure 3 presents the confusion matrices with respect to the shallow and deep classifiers of COPS-CD-0.8, COPS-AV, and DeepCT-CD-0.8 on the SleepAccel dataset. The upper part of Figure 4 illustrates the ground truth sleep stages within a night, and the corresponding normalized ACC and PPG signals over time. The lower part visualizes the activation maps of the shallow and deep classifiers for each method, represented in red and blue, respectively.

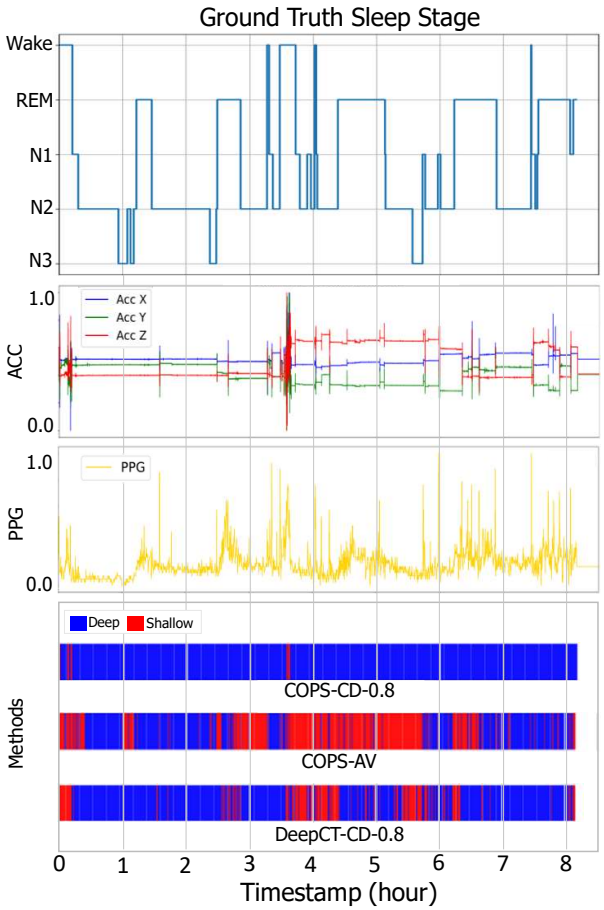


Fig. 4. Visualization of intelligent classifier selection.

The overall accuracy of the shallow classifiers (0.92, 0.78, and 0.92) are generally higher than their corresponding deep classifiers (0.74, 0.65, and 0.68) for COPS-CD-0.8, COPS-AV, and DeepCT-CD-0.8. This phenomenon manifests the effectiveness of the designed confidence-based mechanism CESwitch, as the results of the shallow classifiers are used only when there is sufficient confidence in their predictions. Although the deep classifiers have more layers to learn complex relationships hidden in the features, their accuracy is lower because they only handle the most challenging instances.

Sleep stage N2 has the largest amount of data, making it easier to learn compared to other stages. It results in high accuracy ranging from 0.75 to 0.94 among all classifiers. COPS-AV effectively retains most of the simpler N2 cases in the shallow classifier, thereby enhancing power savings. In contrast, recognizing stage N1 is difficult for all classifiers due to its small representation (only 6.6%) in the dataset. As a result, all shallow classifiers struggle with N1 instances, passing most of them to the deep classifiers.

In Figure 4, both REM and N2 stages exhibit stable ACC signals due to little body movement, but only REM shows greater heart rate variations through PPG signals. The similarity in ACC signals makes CD strategies share comparable probabilities for these two stages, neither of which is high

enough to pass the confidence threshold for using the shallow classifier, resulting in a smaller red area in their activation maps. In contrast, AV strategy operates differently: given these signals, the binary classifier verifies that the shallow and deep classifiers are likely to agree on the prediction. Therefore, AV uses the shallow classifier more often, leading to greater red area and more energy savings. Note that the binary classifier does not directly verify the predicted sleep stages but instead focuses on whether the classifiers will reach a consensus.

Although stage Wake has a smaller amount of data (8.7%) similar to N1, it is easier to identify (at least 62.8% in accuracy) due to the significantly active ACC and PPG signals detected, as illustrated at around timestamps 0 and 3.5 in Figure 4. It is worth noting that, 18.93%, 54.82%, and 37.06% of the wakefulness instances (i.e., stage Wake) are managed by the power-efficient shallow classifiers with respect to COPS-CD-0.8, COPS-AV, and DeepCT-CD-0.8, implying potential power savings during daytime compared to conventional works that used a deep classifier alone.

VI. Conclusion

This paper introduces COPS, a novel confidence-based, power-efficient sleep stage classification framework designed for consumer wearable devices. Unlike traditional methods that utilized a single deep classifier for all cases, COPS additionally conducts a shallow classifier to handle simpler cases, which requires fewer computational costs and thereby saves energy. To distribute input instances to the appropriate classifier, an innovative intelligent mechanism, CESwitch, derives confidence scores to determine which classifier should be activated. Two strategies of CESwitch, Confidence Delegation and Agreement Verification, are proposed to address different scenarios. Therefore, COPS strikes a good balance between power efficiency and classification accuracy. Moreover, both COPS and CESwitch can be easily adapted to existing deep models. Comprehensive experimental results on real-world datasets demonstrate the efficacy, energy efficiency, and versatility of COPS in terms of FLOPs and accuracy.

References

[1] L. S. Dhingra, A. Aminorroaya, E. K. Oikonomou, A. A. Nargesi, F. P. Wilson, H. M. Krumholz, and R. Khera, "Use of Wearable Devices in Individuals With or at Risk for Cardiovascular Disease in the US, 2019 to 2020," *JAMA Network Open*, 2023.

[2] S. H. Friend, G. S. Ginsburg, and R. W. Picard, "Wearable digital health technology," *New England Journal of Medicine*, 2023.

[3] R. Stickgold and M. P. Walker, "Sleep-dependent memory triage: evolving generalization through selective processing," *Nature Neuroscience*, 2013.

[4] L. Xie, H. Kang, Q. Xu, M. J. Chen, Y. Liao, M. Thiyagarajan, J. O'Donnell, D. J. R. Christensen, C. Nicholson, J. J. Iliff, T. Takano, R. Deane, and M. Nedergaard, "Sleep drives metabolite clearance from the adult brain," *Science*, 2013.

[5] M. R. Irwin and M. R. Opp, "Sleep health: Reciprocal regulation of sleep and innate immunity," *Neuropsychopharmacology*, 2017.

[6] D. J. Buysse, C. F. Reynolds, T. H. Monk, S. R. Berman, and D. J. Kupfer, "The pittsburgh sleep quality index: A new instrument for psychiatric practice and research," *Psychiatry Research*, 1989.

[7] R. Dobson, M. Stowell, J. Warren, T. Tane, L. Ni, Y. Gu, J. McCool, and R. Whittaker, "Use of consumer wearables in health research: Issues and considerations," *Journal of Medical Internet Research*, 2023.

[8] M. R. Lujan, I. Perez-Pozuelo, and M. A. Grandner, "Past, present, and future of multisensory wearable technology to monitor sleep and circadian rhythms," *Frontiers in Digital Health*, 2021.

[9] A. Henriksen, M. H. Mikalsen, A. Z. Woldaregay, M. Muzny, G. Hartvigsen, L. A. Hopstock, and S. Grimsgaard, "Using fitness trackers and smartwatches to measure physical activity in research: Analysis of consumer wrist-worn wearables," *Journal of Medical Internet Research*, 2018.

[10] N. Che, T. Zhang, Y. Li, F. Yu, and H. Wang, "Rlsf: Multimodal sleep improvement based reinforcement learning," *IEEE Access*, 2023.

[11] R. Xu, W. Jin, and D. Kim, "Environment optimization scheme based on edge computing using pso for efficient thermal comfort control in resident space," *Actuators*, 2021.

[12] X. Liu and A. G. Richardson, "A system-on-chip for closed-loop optogenetic sleep modulation," in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2021.

[13] A. Nguyen, G. Pogoncheff, B. X. Dong, N. Bui, H. Truong, N. Pham, L. T. Nguyen, H. H. Nguyen, S. Duong-Quy, S. Ha, and T. Vu, "A large-scale study of a sleep tracking and improving device with closed-loop and personalized real-time acoustic stimulation," *ArXiv*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:253370232

[14] C. Iber, S. Ancoli-Israel, A. L. Chesson, and S. F. Quan, "The american academy of sleep medicine (aasm) manual for the scoring of sleep and associated events: Rules, terminology and technical specifications," 2007.

[15] G. Huang, Y. Yuan, G. Cao, and F. Ma, "Accsleepnet: An axis-aware hybrid deep fusion model for sleep stage classification using wrist-worn accelerometer data," in *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2022.

[16] M. Olsen, J. M. Zeitzer, R. N. Richardson, P. Davidenko, P. J. Jennum, H. B. D. Sørensen, and E. J.-M. Mignot, "A flexible deep learning architecture for temporal sleep stage classification using accelerometry and photoplethysmography," *IEEE Transactions on Biomedical Engineering*, 2022.

[17] P. Fonseca, M. Ross, A. Cerny, P. Anderer, F. B. van Meulen, H. Janssen, A. Pijpers, S. Dujardin, P. V. van Hirtum, M. M. van Gilst, and S. Overeem, "A computationally efficient algorithm for wearable sleep staging in clinical populations," *Scientific Reports*.

[18] Y. Liao, C. Zhang, M. Zhang, Z. Wang, and X. Xie, "Lightsleepnet: Design of a personalized portable sleep staging system based on single-channel eeg," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2022.

[19] G. Liu, G. Wei, S. Sun, D. Mao, J. Zhang, D. Zhao, X.-L. Tian, X. Wang, and N. Chen, "Micro sleepnet: efficient deep learning model for mobile terminal real-time sleep staging," *Frontiers in Neuroscience*, 2023.

[20] A. Supratak, H. Dong, C. Wu, and Y. Guo, "Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel eeg," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2017.

[21] Z. Yao and X. Liu, "A cnn-transformer deep learning model for real-time sleep stage classification in an energy-constrained wireless device*," in *2023 11th International IEEE/EMBS Conference on Neural Engineering (NER)*, 2022.

[22] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.

[23] P. Peng, Y. Song, L. Yang, and H. Wei, "Seizure prediction in eeg signals using stft and domain adaptation," *Frontiers in Neuroscience*, 2022.

[24] M. Cao, T. Zhao, Y. Li, W. Zhang, P. Benharash, and R. Ramezani, "Ecg heartbeat classification using deep transfer learning with convolutional neural network and stft technique," *Journal of Physics: Conference Series*, 2023.

[25] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," 2023. [Online]. Available: https://arxiv.org/abs/1606.08415

[26] O. J. Walch, Y. Huang, D. B. Forger, and C. A. Goldstein, "Sleep stage prediction with raw acceleration and photoplethysmography heart rate data derived from a consumer wearable device," *Sleep*.

[27] G.-Q. Zhang, L. Cui, R. Mueller, S. Tao, M. Kim, M. Rueschman, S. Mariani, D. Mobley, and S. Redline, "The national sleep research resource: towards a sleep data commons," *Journal of the American Medical Informatics Association*, 2018.

[28] K. Wang, J. Yang, A. Shetty, and J. Dunn, "Dreamt: Dataset for real-time sleep stage estimation using multisensor wearable technology."