# Joint Classification and Prediction CNN Framework for Automatic Sleep Stage Classification

Huy Phan*, Fernando Andreotti, Navin Cooray, Oliver Y. Chén, and Maarten De Vos



***Abstract*—Sleep staging plays an important role in assessment and treatment of sleep issues. This work proposes a joint classification-and-prediction framework based on convolutional neural networks (CNNs) for automatic sleep staging, and, subsequently, introduces a simple yet efficient CNN architecture to power the framework. Given a single input epoch, the framework jointly determines its label (classification) and those of its neighboring epochs in the contextual output (prediction). While the proposed framework is orthogonal to the widely adopted classification schemes, which takes one or multiple epochs as a contextual input and produces a single classification decision on the target epoch, we demonstrate its advantages in different ways. First, it leverages the dependency among consecutive sleep epochs while surpassing the problems experienced with the common classification schemes. Second, even with a single model, the framework effortlessly produces multiple decisions as in ensemble-of-models methods which are essential in obtaining a good performance. Probabilistic aggregation techniques are then proposed to leverage the availability of multiple decisions. We demonstrate good performance on the Montreal Archive of Sleep Studies (MASS) dataset consisting 200 subjects with an average accuracy of 83.6%. We also show that the proposed framework not only is superior to the baselines based on the common classification schemes but also outperforms existing deep-learning approaches. To our knowledge, this is the first work going beyond the standard single-output classification to consider neural networks with multiple outputs for automatic sleep staging. This framework could provide avenues for further studies of different neural-network architectures for this problem.**

***Index Terms*—sleep stage classification, joint classification and prediction, convolutional neural network, multi-task.**

## I. Introduction

Identifying the sleep stages from overnight Polysomnography (PSG) recordings plays an important role in diagnosing and treating sleep disorders which affects millions of people [1], [2]. More often than not, this task has been done manually by experts via visual inspection which is tedious, time-consuming, and prone to subjective error. Automatic sleep stage classification [3], ideally as good as manual scoring, would help to ease this task tremendously and, subsequently improve home monitoring of sleep disorders [4].

The guiding principle of automatic sleep staging is to split the signal into a sequence of epochs, usually 30 seconds long, and the classification is then performed epoch-by-epoch. In order to recognize a sleep stage at a certain epoch, proper features need to be derived from the signal, like electroencephalography (EEG). Traditionally, many features have been designed based on our understanding about sleep. These hand-crafted features range from time-domain features [2], [5], [6] to frequency-domain features [6]–[9], via features derived from nonlinear processes [8], [10]–[12]. Using these features, the classification goal is often accomplished by conventional machine learning algorithms. Support Vector Machine (SVM) [6], [13], $k$-nearest neighbors ($k$-NN) [7], Random Forests [14], [15] have been commonly employed for this purpose.

The advent of deep learning along with its astonishing progress in numerous domains have stimulated interest in applying them for automatic sleep staging. The power of deep networks is their great capability of automatic feature learning from data, diminishing the reliance on hand-crafted features. Significant progress on different sleep staging benchmark datasets have been reported by these deep learning techniques [16]–[24], mirroring a relentless trend where learned features ultimately outperform and displace long-used hand-crafted features. CNN [25], [26], the cornerstone of deep learning techniques, has been frequently employed for the task [17]–[19]. The weight sharing mechanism at the convolutional

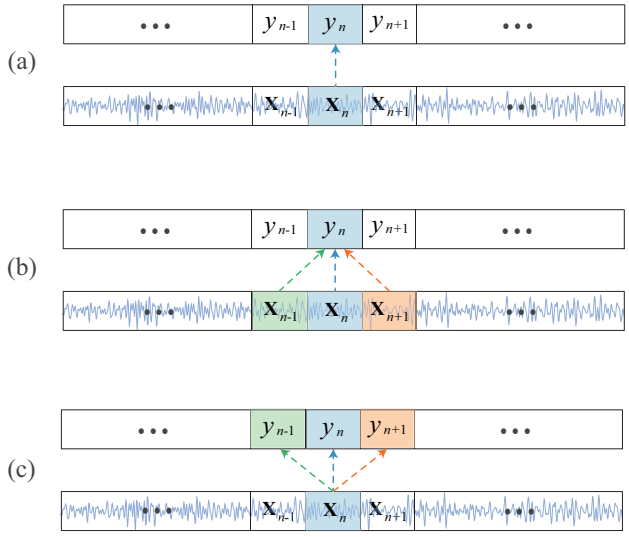


Figure 1: Illustration of (a) the standard classification approach, (b) the common classification approach with the contextual input of three epochs, and (c) the joint classification and prediction with the contextual output of three epochs proposed in this work.

H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos are with the Institute of Biomedical Engineering, University of Oxford, Oxford OX3 7DQ, United Kingdom.

*Corresponding author: huy.phan@eng.ox.ac.uk

layers forces the shift-invariance of the learned features and greatly reduces the model complexity, consequently leading to generalization improvement [25]. Other network variants, such as Deep Belief Networks (DBNs) [27], Auto-encoder [20], Deep Neural Networks (DNNs) [22], have also been explored. Moreover, Recurrent Neural Networks (RNNs), e.g. Long Short-Term Memory (LTSM) [28], which are capable of sequential modelling, have been found efficient in capturing long-term sleep stage transition and usually utilized to complement other network types, such as CNNs [16], [21] and DNNs [22]. Standalone RNNs have also been exploited for learning sequential features of sleep [24]. The classification is usually performed therein by the networks in an end-to-end fashion [17]–[19], however, a separate classifier, such as SVM, can alternatively be used [24], [29].

## II. Motivation and Contributions

### A. Motivation

Sleep is temporal process with slow stage transitions, implying continuity of sleep stages and strong dependency between consecutive epochs [30]–[32]. For instance, out of 228,870 epochs in the entire Montreal Archive of Sleep Studies (MASS) dataset [33] used in this work, $83.3\%$ pairs of adjacent epochs have the same label. Even when two epochs with one epoch away the ratio is still as high as $79.3\%$. This nature of sleep has inspired a widely adopted practice in neural network based sleep staging systems, which makes use of *contextual input* to augment a target epoch by its surrounding epochs (*many-to-one*) in the classification task [16], [17], [19], [21]. Input context size of three and five epochs are often seen [17], [19], [20]. This classification scheme can also be interpreted as an extension of the standard classification setup, i.e. determining the sleep stage corresponding to a single epoch of input signals (*one-to-one*) [23], [24], [34]. Figure 1 (b) illustrates the usage of contextual input of three epochs in comparison with the standard one-to-one classification approach in Figure 1 (a). While multiple-epoch input does not always provide performance gains, as shown in our experiments, it poses a problem of inherent *modelling ambiguity*. That is, when training a network with contextual input, let say three epochs as illustrated in Figure 1 (b), it is questionable whether the network is truly modelling the class distribution of the target epoch at the center or that of the left and right neighbor. In our experiment, such a network (i.e. the many-to-one baseline in Section V-C) achieves an accuracy of $82.1\%$ in determining the labels of the center epochs. However, when aligning the network output with those labels of the left and right neighbor, the accuracy is just marginally lower, reaching $81.1\%$ and $80.8\%$, respectively. Last but not least, the contextual input causes the network's computational complexity increase at a linear scale due to the enlarged input size.

In this work, we formulate sleep staging as a joint classification and prediction, in other words, a *one-to-many* problem, which is an extension from the standard one-to-one classification scheme while being orthogonal to the common many-to-one classification scheme. With this new formulation, given a single target epoch as input, our objective is to simultaneously determine its label (classification) and the labels of its neighboring epochs (prediction) in the *contextual output*, as demonstrated in Figure 1 (c). By classification, we mean determining the label of an epoch given its information. In contrast, prediction implies determining the label of an epoch without knowing its information. The rationale behind this idea is that, given the strong dependency of consecutive epochs, using information of an epoch, we should be able to infer the label of its neighbors. The major benefit of the joint classification and prediction formulation are two-fold. First, with the single-epoch input, the employed model does not experience the modelling ambiguity and the computational overhead induced by the large contextual input as previously discussed. Second, the employed model can effortlessly make available of ensemble of decisions, which is the key in our obtained state-of-the-art performance. Ensemble of models [35], [36], a well-established method to improve the performance of a machine learning algorithm, has been found generalized to automatic sleep staging, evidenced by conventional methods [6], [13], [37] and recently modern deep neural networks [16]. However, building many different models on the same data and then aggregating their predictions are costly and result in cumbersome systems. Opposing to ensemble of models [6], [13], [16], [37], in our joint classification and prediction formulation, the ensemble of decisions is produced with a single multi-task model. Afterwards, an aggregation method can be used to fuse the ensemble of decisions to produce a reliable final decision.

We further proposed a CNN framework to deal with the joint problem. Albeit the proposed framework is generic in the sense that any CNN can fit in, we employ a simple CNN architecture acting on time-frequency image features. The efficiency of this architecture for automatic sleep staging was demonstrated in our previous work [23]. To suit the joint classification and prediction task, we replace the CNN's canonical softmax layer by a *multi-task softmax* layer and introduce *multi-task loss* for network training. Without confusion, we will refer to the proposed framework as multi-task framework, joint classification and prediction framework, and one-to-many framework interchangeably throughout this article.

### B. Contributions

The main contributions of this work are as follows.

(i) We formulate automatic sleep staging as a joint classification and prediction problem. As discussed, the new formulation can avoid the shortcomings of the common classification scheme while holding promise for performance gains.

(ii) A CNN framework is then proposed for the joint problem. To bolster the framework, we present and employ a simple yet efficient CNN coupled with a tailored multi-task softmax layer and multi-task loss to suit joint classification and prediction.

(iii) We further propose two probabilistic aggregation methods, namely additive and multiplicative voting, to leverage ensemble of decisions available in the proposed framework.

(iv) Performance-wise, we empirically demonstrate good performance on the Montreal Archive of Sleep Studies

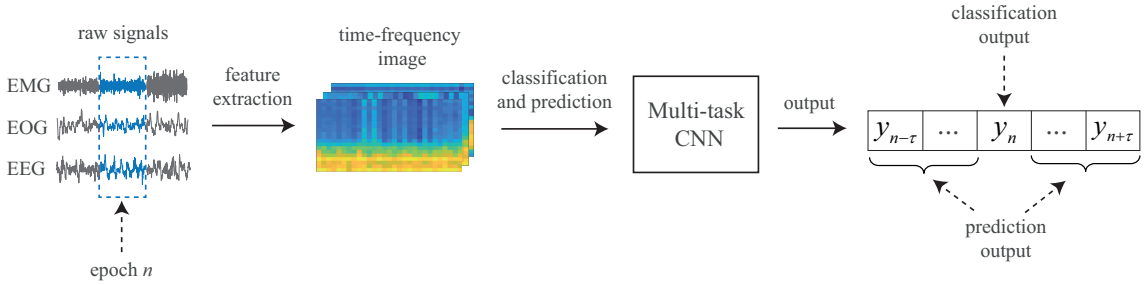


Figure 2: Overview of the proposed joint classification and prediction framework.

(MASS) dataset [33], a large publicly available sleep dataset with 200 subjects.

## III. Montreal Archive of Sleep Studies (MASS) Dataset

We used the public dataset Montreal Archive of Sleep Studies (MASS) in this work and conducted analysis with both unimodal (i.e. single-channel EEG) and multimodal conditions (i.e combinations of EEG, EOG, and EMG channels). It should be noted that even though we selected the typical EEG, EOG, and EMG channels for our analysis, studying different channel combinations should be straightforward with the proposed framework.

MASS comprises whole-night recordings from 200 subjects (97 males and 103 females with an age range of 18-76 years). These recordings were pooled from different hospital-based sleep laboratories. The available cohort 1 was divided into five subsets of recordings, SS1 - SS5. As stated in the seminal work [33], heterogeneity between subsets is expected. Opposing to majority of previous works which targeted only one homogeneous subset of the cohort [21], [22], we experimented with all five subsets as a whole. Each epoch of the recordings was manually labelled by experts into one of five sleep stage {W, N1, N2, N3, and REM} according to AASM standard [30]. We adopted and studied combinations of the C4-A1 EEG, an average EOG (ROC-LOC), and an average EMG (CHIN1-CHIN2) channels in our experiments. The signals, originally sampled at 256 Hz, were downsampled to 100 Hz. Those recordings with 20-second epochs were converted into 30-second ones by including 5-second segments before and after each epoch.

## IV. Joint Classification and Prediction CNN Framework

### A. Overview

The proposed framework, as shown in Fig. 2, is consistent for both unimodal and multimodal input. It can be described in a stage-wise fashion. The raw signals of a certain epoch index $n$ are first transformed into log-power spectra. The spectra are then preprocessed for frequency smoothing and dimension reduction using frequency-domain filter banks. The resulting channel-specific images are then stacked to form a multi-channel time-frequency image, denoted as $\mathbf{X}_n$. Subsequently, a multi-task CNN is exercised on the multi-channel time-frequency image for joint classification and context prediction. The former task is to maximize the conditional probability $P(y_n \,|\, \mathbf{X}_n)$ which characterizes the likelihood of a sleep stage $y_n \in \mathcal{L} = \{1, 2, \ldots, Y\}$, where $\mathcal{L}$ denotes the label set of $Y$ sleep stages. The latter one is to maximize the conditional probabilities $(P(y_{n-\tau} \,|\, \mathbf{X}_n), \ldots, P(y_{n-1} \,|\, \mathbf{X}_n), P(y_{n+1} \,|\, \mathbf{X}_n), \ldots, P(y_{n+\tau} \,|\, \mathbf{X}_n))$ of the neighboring epochs in the output context size of $2\tau + 1$. The labels of the epochs in the output context, where $(y_{n-\tau}, \ldots, y_n, \ldots, y_{n+\tau})$, can be easily obtained by conditional probability maximization.

Formally, under this joint classification and prediction formulation, the CNN is purposed to perform the one-to-many mapping

$$\hat{\mathcal{F}} : \mathbf{X}_n \mapsto (y_{n-\tau}, \ldots, y_n, \ldots, y_{n+\tau}) \in \mathcal{L}^{2\tau+1}. \qquad (1)$$

Note that the order of the epochs in the neighborhood is encoded by the order of the output labels. This formulation is orthogonal to the common classification one with contextual input of size $2\tau+1$, in which a network is supposed to perform the many-to-one mapping

$$\mathcal{F} : (\mathbf{X}_{n-\tau}, \ldots, \mathbf{X}_n, \ldots, \mathbf{X}_{n+\tau}) \mapsto y_n \in \mathcal{L}. \qquad (2)$$

Both formulations (1) and (2) can be interpreted as different extensions of the standard one-to-one classification scheme [23], [24], [34]. They will reduce to this standard one when $\tau = 0$. However, with our joint classification and prediction formulation, at a certain epoch index $n$ there exists an ensemble of exactly $2\tau + 1$ decisions, one classification decision made by itself (i.e. $\mathbf{X}_n$) and $2\tau$ prediction decisions made by its neighbors $(\mathbf{X}_{n-\tau}, \ldots, \mathbf{X}_{n-1}, \mathbf{X}_{n+1}, \ldots, \mathbf{X}_{n+\tau})$. These decisions can be aggregated to form the final decision which is generally better that any individual ones.

### B. Time-Frequency Image Representation

Given a 30-second signal epoch (i.e. EEG, EOG, or EMG), we firstly transform it into a power spectrum using short-time Fourier transform (STFT) with a window size of two seconds and 50% overlap. Hamming window and 256-point Fast Fourier Transform (FFT) are used. The spectrum is then converted to logarithm scale to produce a log-power spectrum image of size $F \times T$, where $F = 129$ and $T = 29$.

For frequency smoothing and dimension reduction, the spectrum is filtered by a frequency-domain filter bank. Any frequency-domain filter bank, such as the regular triangular

one [23], could serve this purpose. However, it is more favorable to learn the filter bank specifically for the task at hand. Our recent works in [23], [24] demonstrated that a filter bank learned by a DNN in a discriminative fashion is more competent than the regular one. The learned filter bank is expected to emphasize the subbands that are more important for the task and attenuate those less important. Hence, we use the DNN-learned filter bank as an off-the-shelf component for preprocessing here. One such filter bank with $M = 20$ filters is learned for each EEG, EOG, and EMG channel. Filtering the log-power spectrum image reduces its size to $M \times T$. When multiple channels are used, we obtain one such time-frequency image for each channel. For generalization, we denote the time-frequency image as $\mathbf{X} \in \mathbb{R}^{P \times M \times T}$ where $P$ denotes the number of channels. $P = 1, 2, 3$ is equivalent to the cases when {EEG}, {EEG, EOG}, and {EEG, EOG, EMG} are employed, respectively.

### C. Multi-Task CNN for Joint Classification and Prediction

Our recent work [23] presented a simple CNN architecture which was shown efficient for sleep staging. We adapt this architecture here by tailoring the last layer, i.e. the multi-task softmax layer, to perform joint classification and prediction. The proposed CNN architecture is illustrated in Figure 3. Opposing to typical deep CNNs [16], [17], [19], [21], the proposed CNN consists only three layers: one over-time convolutional layer, one pooling layer, and one *multi-task softmax* layer. This simple architecture has three main characteristics. First, similar to those in [21], [38], [39], its convolutional layer simultaneously accommodates convolutional kernels with varying sizes, and therefore, is able to learn features at different resolutions. Second, the exploited *1-max* pooling strategy at the pooling layer is arguably more suitable for capturing the *shift-invariance* property of temporal signals than the common subsampling pooling since a particular feature could occur at any temporal position rather than in a local region of the input signal [38]–[40]. Third, opposing to the canonical softmax, the multi-task softmax layer is adapted to suit the joint classification and prediction. Furthermore, the multi-task loss is introduced for network training.

Assume that we obtain a training set $\mathcal{S} = \left\{\left(\mathbf{X}_{n_i}^{(i)}, (\mathbf{y}_{n_i-\tau}^{(i)}, \ldots, \mathbf{y}_{n_i}^{(i)}, \ldots, \mathbf{y}_{n_i+\tau}^{(i)})\right)\right\}_{i=1}^{N}$ of size $N$ from the training data. An epoch $i$ is represented by the multi-channel time-frequency image $\mathbf{X}_{n_i}^{(i)} \in \mathbb{R}^{P \times M \times T}$ as described in IV-B and $n_i$ denotes the corresponding index of the epoch in the original signal. Each epoch $i$ is associated with the sequence of one-hot encoding vectors $(\mathbf{y}_{n_i-\tau}^{(i)}, \ldots, \mathbf{y}_{n_i}^{(i)}, \ldots, \mathbf{y}_{n_i+\tau}^{(i)})$ which represent the sleep stages of the epochs in the context $[n_i - \tau, n_i + \tau]$ of size $2\tau + 1$. We use this training set to train the multi-task CNN for joint classification and context prediction.

*1) Over-Time Convolutional Layer:* Each 3-dimensional filter $\mathbf{w} \in \mathbb{R}^{P \times M \times w}$ of the convolutional layer has the temporal size of $w < T$ while the frequency and channel size entirely cover the frequency and channel dimension of a multi-channel time-frequency image input. The filter is convolved with the input image over time with a stride of 1. ReLU activation [41] is then applied to the feature map.

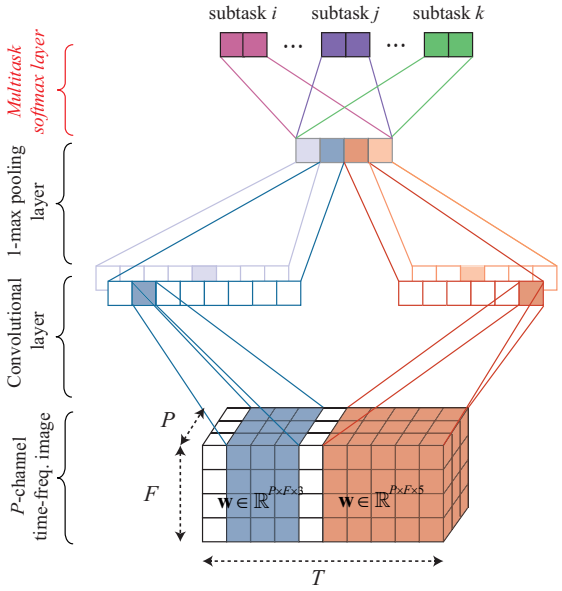


Figure 3: Illustration of the proposed multi-task CNN architecture. The convolution layer of the CNN consists of two filter sets with temporal widths $w = 3$ and $w = 5$. Each filter set has two individual filters. The colors of the output layer indicate different subtasks jointly modelled by the network.

The CNN is designed to have $R$ filter sets with different temporal widths $w$ to be able to capture features at multiple temporal resolutions. Each filter consists of $Q$ different filters of the same temporal width to encourage the CNN to learn multiple complementary features. As a result, the total number of filters is $Q \times R$

*2) 1-Max Pooling Layer:* We employ 1-max pooling function [39], [40] on a feature map produced by convolving a filter over an input image to retain the most prominent feature. Pooling all feature maps of $Q \times R$ filters results in a feature vector of size $Q \times R$.

With the over-time convolution layer coupled with the 1-max pooling layer, the CNN functions as a template learning and matching algorithm. The convolutional filters play the role of time-frequency templates that are tuned to be useful for the task at hand. Convolving a filter through time can be interpreted as template matching operation, resulting in a feature map which indicates how well the template is matched to different parts of the input image. In turn, 1-max pooling retains a single maximum value, i.e. the maximum matching score, of the feature map as the final feature.

*3) Multi-Task Softmax Layer:* Opposing to a classification network which typically uses the canonical softmax layer for classification, we propose a multi-task softmax layer to suit joint classification and prediction. The idea is that the network should be penalized for both misclassification and misprediction on a training example. The classification and prediction errors on a training example $i$ is computed as the

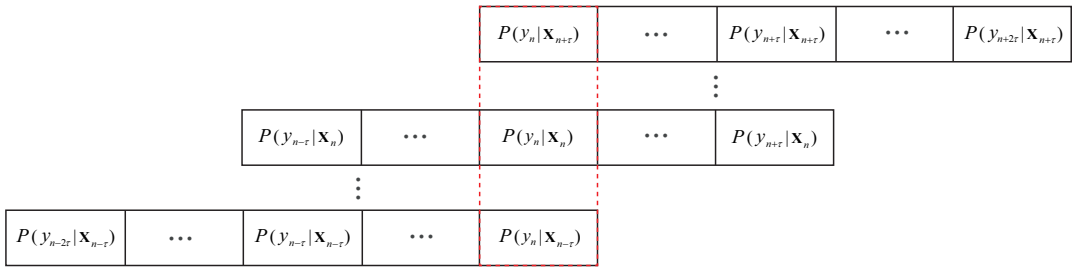


Figure 4: Ensemble of decisions available at the epoch index $n$ made by the epochs $\mathbf{X}_i$ in the neighborhood $[n-\tau, n+\tau]$, i.e. $n-\tau \leq i \leq n+\tau$.

sum of the cross-entropy errors on the individual subtasks:

$$E^{(i)}(\boldsymbol{\theta}) = \sum_{n=n_i-\tau}^{n_i+\tau} \mathbf{y}_n^{(i)} \log\left(\hat{\mathbf{y}}_n^{(i)}(\boldsymbol{\theta})\right), \tag{3}$$

where $\boldsymbol{\theta}$ and $\hat{\mathbf{y}}$ denote the network parameters and the probability distribution outputted by the CNN, respectively.

The network is trained to minimize the multi-task cross-entropy error over $N$ training samples:

$$E(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{i=1}^{N} E^{(i)}(\boldsymbol{\theta}) + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2. \tag{4}$$

Here, $\lambda$ denotes the hyper-parameter that trades off the error terms and the $\ell_2$-norm regularization term. For further regularization, *dropout* [42] is also employed. The network training is performed using the *Adam* optimizer [43].

### D. Ensemble of Decisions and Aggregation

As previously mentioned, one major advantage of the proposed framework is the availability of multiple decisions on a certain epoch even with a single model (the multi-task CNN in this case). Practically, the classification and prediction outputs on a certain epoch may be inconsistent as in ensemble-of-models methods [35], [36], aggregation of these multi-view decisions is necessary to derive a more reliable one. We study two probabilistic aggregation schemes for this purpose: additive and multiplicative voting.

Let $P(y_n \,|\, \mathbf{X}_i)$ denotes the estimated probability output on the sleep stage $y_n \in \mathcal{L}$ at the epoch index $n$ given the epoch $\mathbf{X}_i$ in the neighborhood $[n-\tau, n+\tau]$, i.e. $n-\tau \leq i \leq n+\tau$, as illustrated in Figure 4. The likelihood $P(y_n)$ obtained by additive and multiplicative voting is given by

$$P(y_n) = \frac{1}{2\tau+1}\sum_{i=n-\tau}^{n+\tau} P(y_n \,|\, \mathbf{X}_i), \tag{5}$$

$$P(y_n) = \frac{1}{2\tau+1}\prod_{i=n-\tau}^{n+\tau} P(y_n \,|\, \mathbf{X}_i), \tag{6}$$

respectively. Eventually, the predicted label $\hat{y}_n$ is determined by likelihood maximization:

$$\hat{y}_n = \arg\max_{y_n} P(y_n) \text{ for } y_n \in \mathcal{L}. \tag{7}$$

Table I: Parameters of the proposed CNN.

| Parameter | Value |
|---|---|
| Filter width $w$ | $\{3, 5, 7\}$ |
| Number of filters $Q$ | varied |
| Output context size | 3 |
| Dropout | 0.2 |
| $\lambda$ for regularization | $10^{-3}$ |

Between the two aggregation schemes, the multiplicative one favors likelihoods of categories with consistent decisions and suppresses likelihoods of those categories with diverged decisions stronger than the additive counterpart [44].

## V. Experiments

We aim at achieving several goals in the conducted experiments. Firstly, we empirically prove the feasibility of predicting labels of the neighboring epochs in the output context concurrently with classifying the current one, as a consequence, validate the proposed framework. Secondly, we demonstrate the advantages of the joint classification and prediction (i.e. many-to-one) formulation over the common adopted many-to-one as well as the standard one-to-one classification scheme. Thirdly, we provide performance comparison with various developed baseline systems as well as other deep-learning approaches recently proposed for sleep staging to illustrate the proposed framework's efficiency.

### A. Experimental Setup

We performed 20-fold cross validation on the MASS dataset. At each iteration, 200 subjects were splitted into training, validation, and test set with 180, 10, and 10 subjects, respectively. The sleep staging performance over 20 folds will be reported.

### B. Parameters

The parameters associated with the proposed CNN are given in Table I. We varied the number of convolutional filters $Q$ of the CNN in the set $\{100, 200, 300, 400, 500, 1000\}$ to investigate its influence. Furthermore, we experimented with the output context size of 3 (equivalent to $\tau = 1$). Influence of this parameter will be further discussed in Section VI.

The network implementation was based on *Tensorflow* framework [45]. The network was trained for 200 epochs with

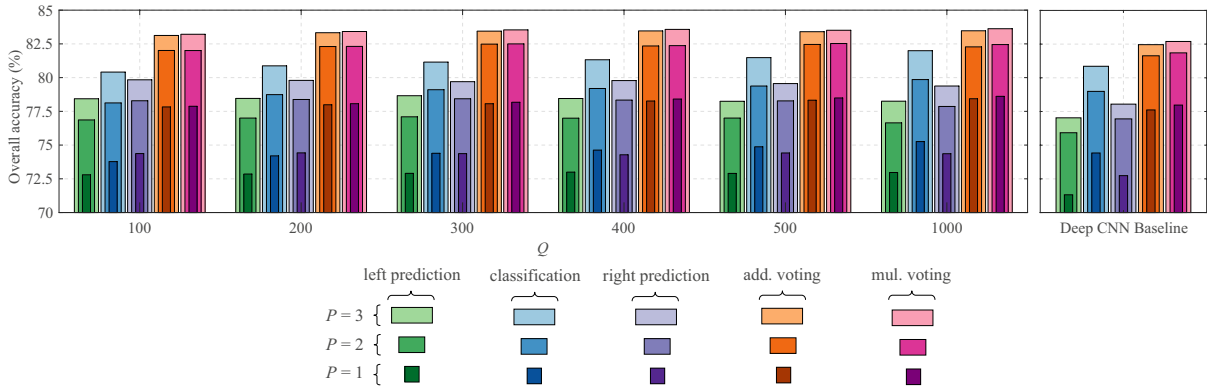


Figure 5: Accuracies of the left prediction subtask, classification subtask, right prediction subtask, multi-task with additive (add.) voting, and multi-task with multiplicative (mul.) voting obtained with an output context size of 3 ($\tau = 1$) and different number of modalities $P$.

Table II: The parameters of the deep CNN baseline.

| Layer | Size | #Fmap | Activation | Dropout |
|---|---|---|---|---|
| conv1 | $3 \times 3$ | 96 | ReLU | - |
| pool1 | $2 \times 1$ | - | - | 0.2 |
| conv2 | $3 \times 3$ | 96 | ReLU | - |
| pool2 | $2 \times 2$ | - | - | 0.2 |
| fc1 | 1024 | - | ReLU | 0.2 |
| fc2 | 1024 | - | ReLU | 0.2 |

a batch size of 200. The learning rate was set to $10^{-4}$ for the *Adam* optimizer. During training, the network which yield the best overall accuracy on the validation set was retained for evaluation. Furthermore, we always randomly generated a data batch to have an equal number of samples for all sleep stages to mitigate the class imbalance issue commonly seen in sleep data.

### C. Baseline Systems

To manifest the advantages offered by the proposed frameworks, we constructed two baseline frameworks for comparison:

- One-to-one: this baseline complies with the standard classification setup, taking a single epoch as input and producing a single decision about its label.
- Many-to-one: this baseline conforms to the commonly adopted scheme with contextual input and outputs a single decision on a target epoch. We fixed the contextual input size to 3, i.e. we augmented a target epoch with two nearest neighbors on its left- and right-hand side.

Both baseline frameworks were designed to maintain common experimental settings as those of the proposed one-to-many framework, i.e. the CNN architecture, the learned filter bank etc. However, it is necessary to use the canonical softmax layer and the standard cross-entropy loss for their classification-only purpose.

We also developed and repeated the experiments with a typical deep CNN architecture as alternative to the proposed CNN described in Section IV-C. This deep CNN baseline consists of 6 layers (2 convolutional layers, 2 subsampling layers, and 2 fully connected layers) with their parameters characterized in Table II. For simplicity, we refer to our proposed CNN as 1-max CNN to distinguish from the deep CNN baseline. With these experiments, our goal is to showcase the generalization of the proposed framework regardless the network base as well as the efficacy of the 1-max CNN in comparison to a typical deep CNN architecture.

### D. Experimental Results

*1) Classification vs prediction accuracy:* In this experiment, we want to empirically validate the proposed framework by demonstrating the feasibility of context prediction. Since we employed the output context size of 3, without confusion, let us refer to the network's subtasks as *classification*, *left prediction*, and *right prediction* which correspond to decisions on the input epoch, its left neighbor, and its right neighbor.

We show in Figure 5 the accuracies of classification, left prediction, and right prediction subtasks obtained by the 1-max CNN (with varying number of convolutional filters $Q$) and the deep CNN baseline with the different number of input modalities $P$. Unlike the classification subtask, the CNNs do not have access to the signal information of the left and right neighboring epochs. As a result, inference for their labels relies solely on their dependency with the input epoch. It can be expected that the accuracies of the left and right prediction subtasks are lower than that of the classification subtask in most of the cases. Nevertheless, overall both CNNs maintain a good accuracy level in prediction relative to the classification accuracy, especially in multimodal cases (e.g. $P = 3$). More specifically, averaging over all $Q$ and $P$, the left and right prediction accuracies of the 1-max CNN are only $2.2\%$ and $1.3\%$ lower than the classification accuracy. Similar patterns can also be seen with the deep CNN baseline with the graceful degradation of $3.3\%$ and $2.2\%$ correspondingly. These results strengthen the assumption about the dependency between neighboring PSG epochs and consolidate the feasibility of joint classification and prediction modelling.

*2) Advantages of the joint classification and prediction:* Figure 5 also highlights the performance improvements ob-

Table III: Performance comparison of different systems developed in this work as well as those in previous works. We marked in bold the figures where the combination of the one-to-many framework and 1-max CNN outperforms all other opponents.

| | $P = 1$ (EEG only) | | | | | $P = 2$ (EEG + EOG) | | | | | $P = 3$ (EEG + EOG + EMG) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acc. | $\kappa$ | MF1 | Sens. | Spec. | Acc. | $\kappa$ | MF1 | Sens. | Spec. | Acc. | $\kappa$ | MF1 | Sens. | Spec. |
| **One-to-many + 1-max CNN** | **78.6** | **0.70** | **70.6** | 71.2 | **94.1** | **82.5** | **0.75** | **76.1** | **75.8** | **95.0** | **83.6** | **0.77** | **77.9** | 77.4 | **95.3** |
| *One-to-one + 1-max CNN* | 75.9 | 0.67 | 69.6 | 71.1 | 93.7 | 80.7 | 0.73 | 74.9 | 75.5 | 94.8 | 82.7 | 0.75 | 77.6 | 77.8 | 95.1 |
| *Many-to-one + 1-max CNN* | 76.3 | 0.67 | 69.8 | 71.3 | 93.8 | 80.9 | 0.73 | 75.1 | 75.5 | 94.8 | 82.1 | 0.75 | 76.6 | 76.9 | 95.0 |
| *One-to-many + deep CNN* | 78.0 | 0.69 | 69.8 | 70.1 | 93.8 | 81.9 | 0.74 | 75.2 | 74.7 | 94.8 | 82.7 | 0.75 | 76.9 | 76.3 | 95.0 |
| *One-to-one + deep CNN* | 74.5 | 0.65 | 68.4 | 70.0 | 93.4 | 79.2 | 0.71 | 73.5 | 74.3 | 94.4 | 81.0 | 0.73 | 76.4 | 77.4 | 94.9 |
| *Many-to-one + deep CNN* | 77.4 | 0.68 | 71.6 | **72.8** | 94.0 | 81.2 | 0.73 | 76.0 | **76.4** | 94.8 | 82.4 | 0.75 | 78.2 | **78.9** | 95.2 |
| Deep CNN [19] (our implementation) | 73.4 | 0.61 | 62.8 | 62.6 | 92.2 | 76.8 | 0.66 | 66.9 | 66.9 | 93.3 | 77.9 | 0.68 | 70.4 | 69.4 | 93.5 |
| DeepSleepNet [21] (our implementation) | 77.9 | 0.68 | 70.0 | 69.2 | 93.7 | 81.1 | 0.73 | 75.4 | 74.7 | 94.6 | 80.7 | 0.73 | 75.8 | 75.5 | 94.5 |

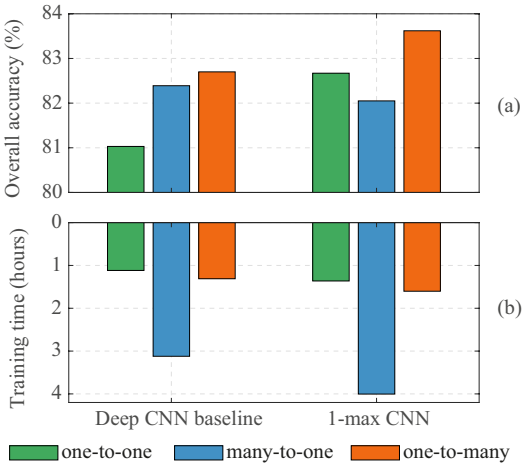


Figure 6: The overall classification accuracy (a) and the amount of training time (b) of the proposed framework in comparison with those of the one-to-one, and many-to-one baselines on the first cross-validation fold. We commonly set $Q = 1000$ and $P = 3$.

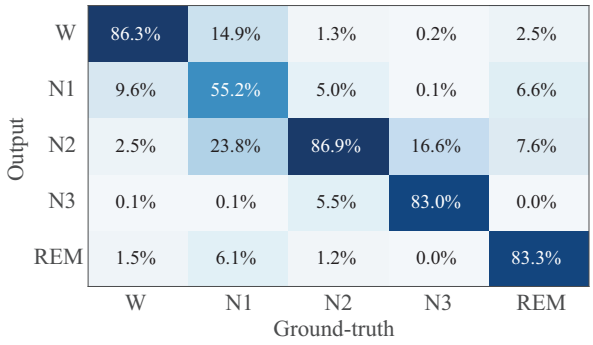


Figure 7: The confusion matrix obtained by the proposed one-to-many framework with the 1-max CNN ($Q = 1000$ and $P = 3$).

tained by the joint classification and prediction framework after the aggregation step in comparison to individual subtasks. Averaging over all $P$ and $Q$, the 1-max CNN with additive and multiplicative voting lead to $4.5\%$ and $4.7\%$ absolute accuracy gains over the classification subtask's accuracy, respectively. Likewise, the deep CNN baseline produces $2.5\%$ and $2.8\%$ absolute gains accordingly. These results emphasize the benefits of leveraging ensemble of decisions (i.e. classification, left prediction, and right prediction). Between two voting schemes, the performance gain of the multiplicative one is slightly better than that of the additive counterpart with a difference around $0.2 - 0.3\%$. It can be explained by the stronger manner of the former compared to the latter [44].

To further shed light on advantages of the proposed framework over the common classification schemes, we further compare its performance and computational complexity with the one-to-one and many-to-one baseline frameworks described in Section V-C. For simplicity, we utilized all available modalities (i.e. $P = 3$) in this experiment and made use of multiplicative-voting aggregation in the proposed framework. Additionally, we set the number of convolutional filters $Q = 1000$ when the 1-max CNN was employed.

Figure 6 depicts the overall accuracy obtained by the three frameworks and their computational complexity in terms of the training time. Note that we only included the training time of the first cross-validation fold as representative and the training time is expected to scale linearly with the amount of training data. Four important points should be noticed from the figure. Firstly, contextual input does not always help as the many-to-one baseline with the 1-max CNN experiences a performance drop of $0.6\%$ absolute albeit it brings up gains with the deep CNN baseline. Secondly, the proposed one-to-many framework consistently upholds its superiority over other counterparts. Using the 1-max CNN as the base, this framework outperforms the one-to-one and many-to-one opponents with $1.0\%$ and $1.6\%$ absolute, respectively. Similar gains $1.7\%$ and $0.3\%$ are achieved when the deep CNN baseline is used. Thirdly, between the network bases, the 1-max CNN surpasses the deep CNN baseline with an improvements $0.9\%$ absolute although its architecture is much simpler. Fourthly, concerning the computational complexity, thrice larger input of the many-to-one baseline roughly triples the training time compared to that of the one-to-one, for instance, 4.0 hours versus 1.36 hours can be seen with the 1-max CNN. Differently, with the training time of 1.6 hours, using the same network, the proposed framework only causes an extra duration as small as

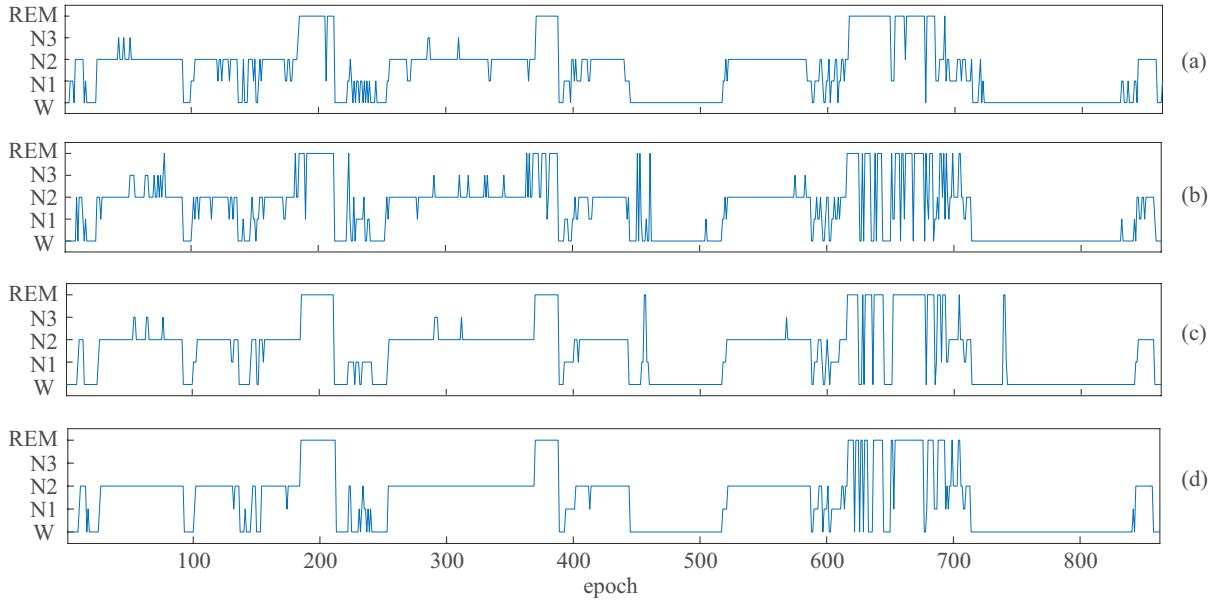


Figure 8: Hypnogram of one subject in the MASS dataset (subject 22 of the subset SS1 [33]): (a) ground-truth, (b) the one-to-one baseline framework's output, (c) the many-to-one baseline framework's output, (d) the proposed one-to-many framework's output. 1-max CNN was commonly used with $Q = 100$ and $P = 3$.

0.2 hours. The training time of the deep CNN baseline also exposes similar patterns.

*3) Performance comparison:* Table III provides a comprehensive performance comparison on the experimental dataset using different metrics, including overall accuracy, kappa index $\kappa$, average specificity, average sensitivity, and average macro F1-score (MF1). The comparison covers all combinations of different frameworks (i.e. the proposed and the baselines) and network bases (i.e. the proposed 1-max CNN and the deep CNN baseline).

To the best of our knowledge, this is the first work evaluating automatic sleep staging performance on entire 200 subjects of the MASS dataset. There exist some recent attempts using deep learning, such as DeepSleepNet in [21] which is the most recent work reporting best results on the MASS dataset as well as Sleep-EDF [46], [47], another popular public sleep dataset. We demonstrated in our recent work [23] that the 1-max CNN with standard classification scheme (i.e. concided with the one-to-one baseline framework) outperformes DeepSleepNet in [21] on Sleep-EDF [46], [47]. However, on MASS, this work only targeted one out of five subsets of the dataset [33] and used different input signals, making a direct comparison inappropriate. To able to compare with previous works while avoiding possible mismatch in experimental setup, we re-implemented DeepSleepNet in [21] for a compatible comparison. Note that we experimented with DeepSleepNet1 (CNN) in [21] here and leave DeepSleepNet2 (CNN combined with RNN for long-term context modelling) for future work. In addition, we also re-implemented the deep CNN proposed in [19] which reported good performance on the Sleep-EDF dataset. With this comprehensive comparison, we also aims at providing a benchmark for future work.

We show the comparison results in Table III. As can be seen, on one hand, the proposed one-to-many framework powered by the 1-max CNN (**one-to-many + 1-max CNN**) outperforms other combinatorial systems developed in this work on both datasets and over different combinations of modalities. There are occasional exceptions where using the 1-max CNN in the baseline frameworks yield marginally better average MF1 and Sensitivity than **one-to-many + 1-max CNN**, such as with $P = 3$, however, **one-to-many + 1-max CNN** remains dominant on other metrics. On the other hand, while performance gains can be seen by many of our developed systems over other deep-network methods in previous work [19], [21], the improvements with **one-to-many + 1-max CNN** are most prominent. For instance, compared to DeepSleepNet [21] a margin of 2.9% on overall accuracy can be seen with $P = 3$. We also conducted experiments on the MASS subset SS3 [33] following a similar experimental setup in [21] and saw similar results. Specifically, **one-to-many + 1-max CNN** achieves 82.5 compared to 81.5% reported in [21].

For completeness, we show confusion matrix obtained with **one-to-many + 1-max CNN** and $P = 3$ in Figure 7. Particularly, as shown in the confusion matrix, we obtain a very good accuracy on N1, which has been proven challenging to be correctly recognized [19], [21], [23], [24] due to its similarities with other stages and generally infrequent. We further demonstrate alignment of ground-truth and system-output hypnograms for one subject of the dataset in Figure 8.

## VI. Discussion

In this section, we investigate into the cause of the proposed framework's performance improvement over the baseline ones. Furthermore, the proposed framework encompasses several influential factors, namely the number of convolutional filters $Q$ of the 1-max CNN, the number of input modalities $P$, and

the output context size. We will discuss and elucidate their effects on the framework's performance.

### A. Investigating the cause of improvement

To accomplish this goal, we divided the dataset into *non-transition* and *transition* set and explored how different frameworks perform on them. The former set is the major one ($83.4\%$ epochs in total) consisting of epochs with the same label as their left and right neighbors. The latter, which is the minor set ($16.6\%$ epochs in total), comprises those epochs at stage transitions, i.e. their labels differ from those of their left/right neighbors or both.

The overall accuracy on these sets are shown in Table IV. On one hand, the downgrading accuracy on the transition set reflects the fact about manual labelling of sleep stages that the accuracy is low near stage transitions [48]. Since a 30-second epoch likely contains the signal information of two transitioning stages while only one label is assigned to such an epoch, up to half of its may not match the assigned label. More often than not, the labels assigned to these epochs are subjective to the scorer. The accuracy of the one-to-one baseline framework on this small subset, which is above the chance level, is likely due to the bias towards the scorer's subjectivity. The chance-level accuracy of the many-to-one and one-to-many frameworks, on the other hand, can be explained by the fact that taking into account the left and right neighboring epochs has balanced the contribution of the two transitioning stages.

Disregarding the ambiguous transition set, the cause of performance improvement turns out depending upon the accuracy on the major non-transition set. As can be seen, the proposed framework outperforms the other two with a gap of $2.7\%$ and $1.3\%$ on this set, respectively. Further investigation on this set reveals a substantial label agreement between the proposed framework and the one-to-one baseline, up to $91.0\%$. However, for the remaining $9.0\%$ epochs on which their labels disagree, the proposed framework yields an accuracy of $60.4\%$, roughly double that obtained by the baseline ($30.5\%$). Analogously, in comparison with the many-to-one baseline, the label agreement is as high as $92.0\%$ whereas an accuracy gap of $15.2\%$ is seen on the dissenting subset with $52.4\%$ of the proposed framework compared to $37.2\%$ of the baseline.

### B. Influence of the number of convolutional filters

In general, more features can be learned by the proposed 1-max CNN with the increasing number of convolutional filters $Q$ and one can expect improvement on the performance. However, influence of $Q$ to the framework's performance is very modest as can be seen from Figure 4. For instance, with $P = 3$ and multiplicative-voting aggregation, using $Q = 1000$ only brings up $0.4\%$ absolute gain in overall accuracy over the case of $Q = 100$ even though the number of filters is ten times larger. The slight influence of the number of filters $Q$ suggests that we can maintain a very good performance even with a very compact network.

Table IV: The overall accuracy of different frameworks on MASS's transition and non-transition subsets. The results are obtained by 1-max CNN base with $Q = 1000$ and $P = 3$.

| | Non-transition (Size 83.4%) | Transition (Size 16.6%) |
|---|---|---|
| One-to-many | 89.5 | 53.3 |
| One-to-one | 86.8 | 62.0 |
| Many-to-one | 88.2 | 51.1 |

### C. Benefits of multimodal input

Single-channel EEG has been found prevalent in literature [6], [19], [21], [23], [24], [49] mainly due to its simplicity. However, apart from brain activities, sleep also involves eye movements and muscular activities at different levels. For instance Rapid Eye Movement (REM) stage usually associates with rapid eye movements and high muscular activities are usually seen during Awake stage. As a result, EOG and EMG are valuable additional sources, complementing EEG in multimodal automatic sleep staging systems [16], [17], [34], [50], [51], not to mention their importance in manual scoring rules [30], [52].

Figure 4 sheds light on the benefit of using EOG and EMG to complement EEG in the proposed framework. Significant improvements on overall accuracy can be seen. Averaging over spectrum of $Q$, compared to the single-channel EEG, coupling EEG and EOG leads to an absolute gain of $4.1\%$ which is further boosted by $1.1\%$ with the compound of EEG, EOG, and EMG.

### D. The trade-off problem with the output context size

It is apparently straightforward to enlarge the output context in the proposed framework. Doing so, we are able to increases cardinality of the decision ensemble which is expected to enhance the performance [35]. However, extending the output context confronts us with a trade-off problem. A large context loosens up the link between the input epoch and the far-away neighbors in the output context. More often than not, this deteriorates the prediction decisions on these epochs and, as a consequence, the quality of individual decisions in the ensemble. The low quality of these prediction decisions may outweigh the benefits of the increased cardinality, worsening the performance instead eventually.

To support our argument, we increased the output context size to 5 (i.e. $\tau = 2$) and repeated the experiment in which we set $Q = 1000$ and $P = 3$ for the 1-max CNN. Figure 9 shows the obtained performance alongside those obtained with the output context size of $\{1, 3\}$ (i.e. $\tau = \{0, 1\}$). Note that, with the context size of 1, the framework is reduced to the one-to-one baseline framework described in Section V-C. Obviously, with the context size of 5 the proposed framework still maintains its superiority over the standard classification setup, however, a graceful degradation compared to the context size of 3 can be observed. Specifically, the accuracies obtained by both additive and multiplicative voting schemes slightly decline by $0.3\%$ and $0.2\%$, respectively.

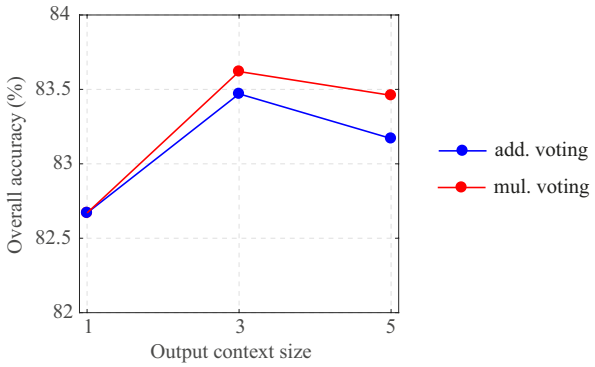


Figure 9: Influence of the output context size to the overall accuracy of the proposed framework. The results obtained with $Q = 1000$ and $P = 3$.

To remedy the weak links between the input epoch and far-away epochs, one possibility would be to combine multiple epochs into the input to form the contextual input. In addition, it would be worth investigating incorporation of long-term context (i.e. in order of dozens of epochs), for example using RNNs as in [16], [21]. However, a detailed study of the proposed frame work in these many-to-many settings is out of the scope of this article and left to future investigation.

## VII. Conclusions

This work has introduce a joint classification and prediction formulation and proposed a multi-task CNN framework for automatic sleep staging. Motivated by the dependency nature of sleep epochs, the framework's purpose is to perform jointly classification of an input epoch and prediction the labels of its neighboring epochs in the context output. While being orthogonal to the widely adopted many-to-one classification scheme relying on contextual input, we argued that the proposed framework could avoid the shortcomings experienced by this scheme, such as the inherent modelling ambiguity and the induced computational overhead due to large contextual input. More importantly, due to multitasking, the framework is able to effortlessly make available of multiple decisions on a certain epoch which afterwards form the reliable final decision via aggregation. We demonstrated the generalization of the framework on the publicly available MASS dataset.

## Acknowledgement

The research was supported by the NIHR Oxford Biomedical Research Centre and Wellcome Trust under Grant 098461/Z/12/Z.

## References

[1] A. C. Krieger, Ed., *Social and Economic Dimensions of Sleep Disorders, An Issue of Sleep Medicine Clinics*. Elsevier, 2017.
[2] S. J. Redmond and C. Heneghan, "Cardiorespiratory-based sleep staging in subjects with obstructive sleep apnea," *IEEE Trans. Biomedical Engineering*, vol. 53, pp. 485–496, 2006.
[3] K. A. I. Aboalayon *et al.*, "Sleep stage classification using EEG signal analysis: A comprehensive survey and new investigation," *Entropy*, vol. 18, no. 9, p. 272, 2016.
[4] J. M. Kelly, R. E. Strecker, and M. T. Bianchi, "Recent developments in home sleep-monitoring devices," *ISRN Neurology*, vol. 2012, 2012.
[5] A. Krakovská and K. Mezeiová, "Automatic sleep scoring: A search for an optimal combination of measures," *Artificial Intelligence in Medicine*, vol. 53, no. 1, pp. 25–33, 2011.
[6] B. Koley and D. Dey, "An ensemble system for automatic sleep stage classification using single channel EEG signal," *Computers in Biology and Medicine*, vol. 42, no. 12, pp. 1186–95, 2012.
[7] H. Phan *et al.*, "Metric learning for automatic sleep stage classification," in *Proc. EMBC*, 2013, pp. 5025–5028.
[8] K. Susmakova and A. Krakovska, "Discrimination ability of individual measures used in sleep stages classification," *Artificial Intelligence in Medicine*, vol. 44, pp. 261–277, 2008.
[9] J. Fell *et al.*, "Discrimination of sleep stages: a comparison between spectral and nonlinear eeg measures," *Electroencephalography and Clinical Neurophysiology*, vol. 98, no. 5, pp. 401–10, 1996.
[10] D. J. Kim *et al.*, "An estimation of the first positive Lyapunov exponent of the EEG in patients with schizophrenia," *Psychiatry Research*, vol. 98, no. 3, pp. 177–89, 2000.
[11] J. M. Lee *et al.*, "Detrended fluctuation analysis of EEG in sleep apnea using MIT/BIH polysomnography data," *Computers in Biology and Medicine*, vol. 32, pp. 37–47, 2012.
[12] X.-S. Zhang, R. J. Roy, and E. W. Jensen, "EEG complexity as a measure of depth of anesthesia for patients," *IEEE Trans. Biomedical Engineering*, vol. 48, no. 12, pp. 1424–1433, 2001.
[13] E. Alickovic and A. Subasi, "Ensemble SVM method for automatic sleep stage classification," *IEEE Trans. on Instrumentation and Measurement*, 2018.
[14] P. Memar and F. Faradji, "A novel multi-class EEG-based sleep stage classification system," *IEEE Trans. Neural Systems and Rehabilitation Engineering*, vol. 6, no. 1, pp. 84–95, 2018.
[15] R. Boostania, F. Karimzadeha, and M. Nami, "A comparative review on sleep stage classification methods in patients and healthy individuals," *Comput. Methods Programs Biomed.*, pp. 77–91, 2017.
[16] J. B. Stephansen *et al.*, "The use of neural networks in the analysis of sleep stages and the diagnosis of narcolepsy," *arXiv:1710.02094*, 2017.
[17] K. Mikkelsen and M. De Vos, "Personalizing deep learning models for automatic sleep staging," *arXiv:1801.02645*, 2018.
[18] J. Zhang and Y. Wu, "A new method for automatic sleep stage classification," *IEEE Trans. on Biomedical Circuits and Systems*, vol. 11, no. 5, pp. 1097–1110, 2017.
[19] O. Tsinalis *et al.*, "Automatic sleep stage scoring with single-channel EEG using convolutional neural networks," *arXiv:1610.01683*, 2016.
[20] O. Tsinalis, P. M. Matthews, and Y. Guo, "Automatic sleep stage scoring using time-frequency analysis and stacked sparse autoencoders," *Annals of Biomedical Engineering*, vol. 44, no. 5, pp. 1587–1597, 2016.
[21] A. Supratak *et al.*, "Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel eeg," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 1998–2008, 2017.
[22] H. Dong *et al.*, "Mixed neural network approach for temporal sleep stage classification," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, 2017.
[23] H. Phan *et al.*, "Dnn filter bank improves 1-max pooling cnn for automatic sleep stage classification," in *EMBC 2018*, 2018, (accepted).
[24] ——, "Automatic sleep stage classification: Learning sequential features with attention-based recurrent neural networks," in *EMBC 2018*, 2018, (accepted).
[25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
[26] Y. Lecun, *Connectionism in perspective*. Elsevier, 1989, ch. Generalization and network design strategies.
[27] M. Längkvist, L. Karlsson, and A. Loutfi, "Sleep stage classification using unsupervised feature learning," *Advances in Artificial Neural Systems*, vol. 2012, pp. 1–9, 2012.
[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[29] A. H. Ansari *et al.*, "Neonatal seizure detection using deep convolutional neural networks," *International Journal of Neural Systems*, vol. 28, no. 0, p. 1850011, 2018.
[30] C. Iber *et al.*, "The AASM manual for the scoring of sleep and associated events: Rules, terminology and technical specifications," *American Academy of Sleep Medicine*, 2007.
[31] S.-F. Liang *et al.*, "A rule-based automatic sleep staging method," in *Proc. EBMC*, 2011, pp. 6067–6070.
[32] T. Sousa *et al.*, "A two-step automatic sleep stage classification method with dubious range detection," *Computers in Biology and Medicine*, vol. 59, pp. 42–53, 2015.

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

[34] F. Andreotti *et al.*, "Multichannel sleep stage classification and transfer learning using convolutional neural networks," in *EMBC 2018*, 2018, (accepted).

[35] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv:1503.02531*, 2015.

[36] T. G. Dietterich, *Multiple classifier systems*. Springer, 2000, ch. Ensemble methods in machine learning, pp. 1–15.

[37] A. R. Hassan and M. I. H. Bhuiyan, "Automatic sleep scoring using statistical features in the EMD domain and ensemble methods," *Biocybernetics and Biomedical Engineering*, vol. 36, no. 1, pp. 248–255, 2016.

[38] H. Phan *et al.*, "Improved audio scene classification based on label-tree embeddings and convolutional neural networks," *IEEE/ACM Trans. on Acoustics Speech and Signal Processing*, vol. 25, no. 6, pp. 1278–1290, 2017.

[39] ——, "Robust audio event recognition with 1-max pooling convolutional neural networks," in *Proc. INTERSPEECH*, 2016, pp. 3653–3657.

[40] Y. Kim, "Convolutional neural networks for setence classification," in *Proc. EMNLP*, 2014, pp. 1746–1751.

[41] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machine," in *Proc. ICML 2010*, 2010, pp. 807–81.

[42] N. Srivastava *et al.*, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research (JMLR)*, vol. 15, pp. 1929–1958, 2014.

[43] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," in *Proc. ICLR*, no. 1—13, 2015.

[44] H. Phan *et al.*, "Audio scene classification with deep recurrent neural networks," in *Proc. INTERSPEECH*, 2017, pp. 3043–3047.

[45] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv:1603.04467*, 2016.

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

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

[48] R. S. Rosenberg and S. Van Hout, "The american academy of sleep medicine inter-scorer reliability program: Respiratory events," *Journal of Clinical Sleep Medicine*, vol. 10, pp. 447–454, 2014.

[49] C.-E. Kuo and S.-F. Liang, "Automatic stage scoring of single-channel sleep EEG based on multiscale permutation entropy," in *Proc. BioCAS*, 2011, pp. 448–451.

[50] T. Lajnef *et al.*, "Learning machines and sleeping brains: Automatic sleep stage classification using decision-tree multi-class support vector machines," *Journal of Neuroscience Methods*, vol. 250, pp. 94–105, 2015.

[51] C. S. Huang *et al.*, "Knowledge-based identification of sleep stages based on two forehead electroencephalogram channels," *Frontiers in Neuroscience, vol. 8, p. 263, 2014*, vol. 8, p. 263, 2014.

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