ABSTRACT
Neural networks often obtain sub-optimal representations during training, which degrade robustness as well as classification performances. This is a severe problem in applying deep learning to bio-medical domains, since models are vulnerable to being harmed by irregularities and scarcities in data. In this study, we propose a pre-training technique that handles this challenge in sleep staging tasks. Inspired by conventional methods that experienced physicians have used to classify sleep states from the existence of characteristic waveform shapes, or template patterns, our method introduces a cosine similarity based convolutional neural network to extract representative waveforms from training data. Afterwards, these features guide a model to construct representations based on template patterns. Through extensive experiments, we demonstrated that guiding a neural network with template patterns is an effective approach for sleep staging, since (1) classification performances are significantly enhanced and (2) robustness in several aspects are improved. Last but not least, interpretations on models showed that notable features exploited by trained experts are correctly addressed during prediction in the proposed method.

KEYWORDS
Sleep Staging, Neural Network Pre-training, Model Robustness, Model Interpretation

1 INTRODUCTION
Identifying periodic changes in sleep stages is an important task for sleep disorder comprehension [40, 42], since clinical diagnosis on patients are inferred from sleep stages. Physiological activities, which are in form of electrical signals gathered via bio-medical sensors like electroencephalography (EEG) and electrooculography (EOG), are inspected in order to map each sleep period to a set of meaningful sleep stages [33, 39]. This process can be accomplished by trained physicians with expertise and manual annotation of every single sleep period from biomedical signals requires huge amount of time. Hence, a number of researches have been proposed to automate this process. Among these, deep learning techniques have shown promising results, where notable achievements were made from models with convolutional neural networks (CNN) [33] and recurrent neural networks (RNN) [39]. These approaches have led recent improvements in sleep staging with state-of-the-art performances achieved in an end-to-end manner.

Despite these successes, however, there remain important challenges regarding deep learning techniques. Recent studies reported that neural networks often acquire sub-optimal representations during training [5, 9]. In addition, neural networks are more likely to fail in achieving generalizable representations when data is scarce or irregular [19]. These challenges become more severe in bio-medical domains like sleep staging where data acquisition process is time-consuming and costly. To overcome these challenges, a number of works were proposed with components that properly guide neural

| Stage | Description [26, 40] |
|-------|----------------------|
| W     | Wakefulness. Patterns characterized by high frequency waves are observed: alpha (8-13Hz) and beta (13-25Hz). |
| N1    | Short, transitional phase that occurs when sleep begins. Low-voltage, fast EEG activities are observed: theta (4-7Hz) waves. Sometimes, sharp waves are generated. |
| N2    | Intermediate sleep stage which accounts for 50% of sleep. Theta waves and quick bursts of neuronal activities occur: Sleep Spindles (12-15Hz) and K-complexes. |
| N3    | Deep sleep phase. Extremely slow waveforms with high amplitudes are generated: delta (1-4Hz) or slower waves. |
| REM   | Phase where brains become active. Mixed signals of alpha and theta waves are observed. |
networks to obtain desirable representations. Especially, integration of domain knowledge into deep learning has attracted many researchers [15, 23, 25]. These works successfully improved performances of neural networks on bio-medical domains by introducing proper guidance derived from domain knowledge. These previous works suggest that making a neural network focus on significant features in data could be effective.

This work is proposed with similar regards. Based on conven- tions that physicians have relied on occurrences and relations between characteristic waveform shapes when annotating EEG recordings for sleep stage scoring [40], the proposed work extract such distinctive patterns. Then, these characteristic features are used in pre-training the first convolutional layer of neural networks in order to enable neural networks to fully employ distinctive patterns in recordings. In this paper, we refer these patterns as Template patterns. However, defining the exact numerical forms of such signals are not available at the moment, since characteristics of EEG recordings vary significantly across datasets depending on recording channels and patients/experimental conditions. Thus, novel methodology that automatically extracts templates from given datasets should be considered.

In this paper, we propose TRIER, Template-guided neural networks for Robust and Interpretable Sleep Staging Identification from EEG recordings. TRIER is proposed to extract template patterns in recordings and to exploit them during prediction. Here, in order to locate and extract template patterns, we propose a novel convolutional layer. Features in this layer are computed using cosine similarity [24] in place of dot product to correctly reflect waveform shapes regardless of amplitude. Learned filters from the layers are used as pre-trained templates on target neural networks. With the proposed approach, models are capable of obtaining representations built on template patterns, which might be much more generalizable for unseen data. Through extensive experiments, we demonstrated that the proposed method improved baseline architectures in several aspects. First, classification performances are significantly enhanced on several sleep staging datasets. Second, models become robust in various aspects: confidence calibration and robustness to dataset sizes and gaussian noises. Previous works found that improvements in robustness are one of the main earn- ings that pre-training can bring [14] and our results on robustness indicate that the proposed method correctly obtained these utilities from pre-training. Furthermore, we believe such improvements in robustness are important in fostering neural networks to real-world sleep staging tasks. Finally, with interpretation on learned filters and model behaviors, we empirically show that models trained with the proposed method correctly utilized important features in EEG recordings. Such inspection demonstrates that TRIER is much more interpretable for real-world application, since model successfully revealed significant features referenced by experts.

2 RELATED WORKS

2.1 Sleep Staging Algorithms

Previous studies for developing automated sleep staging algorithms counted on manually curated features from EEG recordings. Hand-crafted features were mainly adopted to construct meaningful representations. For instance, specific form of wavelet transformation, which exploits energy level of well-known frequency bands of EEG recordings, was introduced to build effective representations for classification [30]. Another work integrated time-frequency domain information into non-linear features derived from prior knowledge in EEG signals [18]. Novel form of filters in corporation with previously investigated statistical features in EEG recordings were proposed to extract features from the designated frequency bands [1]. These approaches properly constructed representations for EEG recordings based on domain knowledge on the signal. However, they have major shortcomings that they require manual engineering of parameters to find optimal configurations. Recent approaches on sleep staging employed deep learning model. With deep learning, features were constructed automatically with neural networks in an end-to-end manner. Furthermore, these models surpassed the performances of the manually curated models. In these models, convolutional layers are generally employed in feature extracting layers to learn local patterns in data [33, 38]. Some works further exploited recurrent neural networks after the convolutional feature extractors to model temporal relations between sleep stages [27, 39]. Unlike these works where deep learning is just employed in an end-to-end manner, our work further improved baseline architectures by properly guiding models with desired features.

2.2 Confidence Calibration

Besides classification accuracies, several aspects of neural networks arouse as important research problems. A number of recent works pointed out that neural networks are tend to behave in undesir- able ways. First of all, modern neural networks are known to be over-confident even for wrong predictions [12, 20]. This is a severe problem in mass adoption of deep learning, since people cannot reliably trust the outputs of models. Second, neural networks are not robust against small changes in inputs. Recent works reported that trained models are usually prone to produce wrong predic- tions even for imperceptible perturbations [7, 41]. Considering irregularities in data and consequences of model predictions, these vulnerabilities should be taken into consideration when applying neural networks to bio-medical domains. Researchers are starting to work on model robustness in these perspectives. In the analysis of electrocardiogram (ECG), MINA [15] addressed model’s robustness against distortion and perturbations on signals. For sleep staging, REST [8] introduced novel loss terms for robustness against various forms of noises. Our work also considers robustness of models on gaussian noises similar to these works. However, this is the first work to bring confidence calibration issues to sleep staging tasks.

2.3 Guidance on Neural Network Training

Recent findings pointed out that neural networks could acquire undesirable representations, which do not work out robustly. For example, classifiers on image datasets rely heavily on biases from textures other than shape information, which is against our in- tuition [5, 9]. To cope with these limitations, various techniques, which are based on novel data synthesis [9] or representation decorrelating techniques [43], were proposed to guide models to learn desirable representations from human’s perspectives. For analysis of time-series waveforms, representations from the first layer of convolutional layers are often tackled. SincNet [35] parameterized
the first convolutional layer to a sinc function that is designed to operate as a band-pass filter. Wavelet decomposition network defined the first layer to work as wavelet decomposition when analyzing time-series data [44]. Our work is similar to these works as we configured the first convolutional layer with template signals.

In bio-medical domains, proper representations are often derived from domain knowledge. Previous study employed attention mechanism to integrate beat information from three different levels, which are referenced by medical knowledge in analyzing ECG signals [15]. Other work introduced auxiliary task of locating key-features like P-peaks and R-peaks during training in order to guide neural networks [23]. These works further utilized these features when visualizing evidences used during prediction. To the extent of our knowledge, this is the first work to introduce proper guidance to models, which performs sleep staging in the way that real-world experts are annotating EEG recordings.

3 METHODS
Utilizing prior knowledge from domain fields might be helpful as shown by a number of previous works [15, 23]. In this work, we provide target neural networks with prior knowledge that experts used to locate specific waveform patterns in EEG recordings and inspect relations between them. Proposed strategy is to pre-train the first convolutional filters with such template patterns; thus significant parts in data are first recognized. Afterwards, overall representations are constructed based on them. During pre-training phase, we introduce a novel form of neural network in which cosine similarity based convolutional layer and 1-max pooling architecture are deployed in order to effectively locate template patterns from data. Notations for defining the method is listed in Table 2.

3.1 1-max Pooling Layer Architecture for Extracting Significant Patterns
Learned filters in the first convolutional layer of neural networks often lack distinguishable features when analyzing raw time-series recordings. Previous studies reported seemingly noise-like shapes in the learned filters [21, 35]. We believe such representations might not be template patterns in data. Thus, a novel approach is proposed.

Table 2: Notations for Describing the Proposed Approach.

| Symbol | Description |
|--------|-------------|
| x      | Input to the neural network (EEG recording) |
| w      | Weight vector of a convolutional filter |
| o      | Output vector from a convolutional filter |
| w<sup>k</sup> | k<sup>th</sup> filter vector of a convolutional layer |
| o<sup>k</sup> | k<sup>th</sup> channel output vector from a convolutional layer |
| E      | Loss value calculated from the prediction and label |
| L      | Length of a convolutional filter |
| i<sup>∗</sup>, j<sup>∗</sup> | Index of a convolutional output vector that has maximum activation value |
| v<sub>i</sub> | Scalar value at the i<sup>th</sup> index of a vector v |
| v<sub>[i:j]</sub> | Segment of a vector v from index i to j |
| | L<sub>2</sub>-norm of a a vector v |

Motivation. In general 1D convolutional layer, derivatives with regard to a weight vector w can be written as

\[
\frac{\partial E}{\partial w} = \sum_{i=1}^{l} \frac{\partial E}{\partial o_i} x_{[i:i+l-1]}, \quad (1)
\]

with weight values after update

\[
w^{new} = w^{old} - \alpha \sum_{i=1}^{l} \frac{\partial E}{\partial o_i} x_{[i:i+l-1]}.
\]

Applying Fourier Transform to each side of equation 2, we can decompose the gradient value in terms of frequency components. By the linearity of Fourier Transform, weight updating process becomes identical to adding frequency components of each input segment to current filters, weighted by gradient values. As a consequence, resulting summation becomes noisy, since total gradient values might be intermingled by various frequency components, leading to indistinguishable shapes.

Proposed Approach. We exploited 1-max pooling layer to construct distinctive template representations on each of convolutional filters. With 1-max pooling layer, gradients regarding output values, \(\frac{\partial E}{\partial o_i}\), become zero except for the index \(i^*\) with maximum output value. Now, weight update equation will be simplified as:

\[
w^{new} = w^{old} - \alpha \frac{\partial E}{\partial o_{i^*}} x_{i^*+l-1}.
\]

Here, only the waveform patterns in the input segment that is regarded as the most important patterns will be reflected to parameter update during training. Actually, convolutional layer with one-max pooling has already been proposed to learn significant patterns in data. For DNA sequences and protein sequences, convolution with 1-max pooling successfully captured significant patterns in input sequences [2, 36].

We employed 1-max pooling architecture in a similar manner to extract template patterns. However, in case of bio-medical recordings, amplitudes of recordings vary a lot even within the same recording. Affected by amplitudes of input values, normal convolutional operator might not find the location of important features properly. For this reason, additional approach that can effectively handle amplitude changes should be introduced.

3.2 Cosine Similarity Convolution for Addressing Varying Amplitudes
As illustrated in the work of Lhermitte et al. [22], several feature extraction methodologies are highly vulnerable to amplitude changes in time-series recordings. Since dot products in neural networks are vulnerable to amplitude changes [24], mere application of dot products for bio-medical data might degrade the correctness of identified template signals.

Proposed Approach. To handle this issue, the proposed method extends a general 1D convolutional layer by substituting dot products between convolutional filters and input segments with cosine similarity calculations. On such architecture, output feature map can be calculated as:

\[
o_{i^*}^{k} = \frac{w^{k} \cdot x_{[i^*+l-1]}}{|w^{k}| \cdot |x_{[i^*+l-1]}|}.
\]
Furthermore, using the derivatives of cosine similarity in the previous work [24], the derivative of error with regard to positions in convolutional filter weights can be written as:

$$\frac{\partial E}{\partial w^k_i} = \sum_j \frac{\partial E}{\partial x_j} \left( \frac{x_j}{|x[j,i+1]|} - w^k_{ij} \frac{x_{[j,i+1]}}{|w^k||x[j,i+1]|} \right).$$

To simplify the calculation, norms of each convolutional filter are normalized to one in our implementation. Thus, resulting forward and backward operation in cosine similarity convolution can be simplified as:

$$o^k_i = \frac{w^k \cdot x[i,j+1]}{|x[i,j+1]|}$$

$$\frac{\partial E}{\partial w^k_i} = \sum_j \frac{\partial E}{\partial x_j} \frac{x_{[j,i+1]}}{|x[j,i+1]|}.$$  

With 1-max pooling layer introduced before, gradient values in the proposed method are further simplified as:

$$\frac{\partial E}{\partial w^k_i} = \frac{\partial E}{\partial o^k_i} \frac{x_{[j,i+1]}}{|x[j,i+1]|}.$$  

In the above equation, gradient values will only reflect the waveform shapes of the significant segments regardless of amplitude changes in data. Thus, neural networks will better locate the template patterns in data and update their parameters from them.

### 3.3 Pre-training Using Template Patterns

With above-mentioned two components, cosine similarity convolutional layer and 1-max pooling architecture, each convolutional filter will learn desired template patterns from data. Weights in these filters will be used in initializing the first layer of convolutional filters of target architectures. With template patterns acquired with pre-training, models will more easily locate significant features, which are of more distinguishable and interpretable shapes, and learn to utilize those signals in classifying EEG recordings.

### 4 EXPERIMENTAL SETTING

#### 4.1 Dataset and Pre-processing

Models were evaluated on datasets from the previous sleep studies [33, 34, 39]. For all datasets, sleep stages were labeled according to American Academy of Sleep Medicine (AASM), one of the world-standard sleep scoring systems [4]. Following preceding works [33, 39], we only considered five classes (Wake, N1, N2, N3, REM) that are closely related to sleep stages. Except for Sleep-EDF datasets, we re-scaled EEG recordings to have interquartile range (Q3 - Q1) of 1 and median of 0 [33]. For Sleep-EDF dataset, we used raw recordings to reproduce previous results [27, 39]. Unlike the previous work [33], however, we did not zero out extreme values in data. Summaries on employed datasets are as follows.

- **Sleep-EDF Dataset** [11, 16] is a representative database for sleep studies. Fpz-Cz channel from healthy subjects was used in experiments. Two versions of the dataset were used: EDF-2013 (39 recordings) and EDF-2018 (153 recordings).
- **ISRUC-Sleep Dataset** [17] is provided from Hospital of Coimbra University. We experimented on the first (100 subjects) and the third group (10 subjects). C4-A1 channel was used.
- **CAP Dataset** [42] is a sleep EEG dataset containing 108 recordings. C4-A1 or C3-A2 channel was used.
- **SVUH-UCD Dataset** [11] is a set of recordings from 25 subjects provided by St. Vincent’s University Hospital and University College Dublin. EEG channel C3-A2 was used and two recordings with unknown labels were excluded.

#### 4.2 Neural Network Architectures

We experimented on several architectures to demonstrate that our method can be generally applied to sleep staging. In implementing the architectures that will be pre-trained by the proposed method, we slightly modified the structures when first convolutional layer has large strides. For these architectures, striding sizes were changed to 1, since templates cannot be utilized with large strides. Instead, window sizes of max pooling layer that comes right after the first convolutional layer were increased by the same factor. Hence, total number of parameters in the architecture is unchanged and we believe fairness of comparisons is still valid. We give summaries of architectures as follows.

- **Supratak - LSTM** (DeepSleepNet) [39] is a widely referenced architecture consisting of two-branch convolutional layers with different filter sizes (400 and 50), and Long Short-Term Memory (LSTM) layers [27, 33].
- **Supratak - CNN** [39] is a classifier on CNN layers of DeepSleepNet proposed for pre-training phase in original work.
- **Sors** [38] is a 12-layer CNN architecture for sleep staging. This model was used as a baseline in the recent work [8].
- **Dai** [6] is a set of CNNs for audio classification. It has also been referenced in sleep staging [3]. Among four models, 5-layer and 11-layer architectures were evaluated.

#### 4.3 Evaluation on Confidence Calibration

Confidence calibration refers to concordance between model’s output confidence and likeliness of correct prediction [12]. From the perspective of confidence calibration, it is highly desirable to get $N \cdot p$ correct predictions from $N$ predictions with confidence probability of $p$. Such models are deemed as well-calibrated models. We primarily regard maximum values from Softmax Probabilities on model’s output as confidence values [12, 13].

In evaluating the confidence calibration, we used Expected Calibration Error (ECE) [28]. This metric approximates miscalibration rates with differences between confidences and actual accuracies of each probabilistic bin. This can be formulated as:

$$ECE = \sum_{m=1}^{M} \left( \frac{1}{n} \frac{|B_m|}{n} \right) \left| \text{acc}(B_m) - \text{conf}(B_m) \right|,$$

with $M$ denoting the number of bins. To further inspect the calibration of confidences, we provide Reliability Diagram [29]. This diagram visualizes the gap between likeliness of correct predictions and desirable accuracies for each confidence interval. On this diagram, when average accuracy is smaller than the desirable accuracy...
for some intervals, we can infer that model is over-confident about their predictions.

4.4 Interpretation on Model

Interpretation on Template Patterns
To investigate the constructed templates and how they were utilized during prediction, we interpreted learned representations in convolutional filters. Especially, Sleep Spindles and K-complexes are well-known template patterns in EEG recordings during sleep stage 2. We employed DETOKS algorithm [31], which is proposed for detecting Sleep Spindles and K-complexes from raw EEG signals. Even though such detection algorithms are vulnerable to generate false positives and are incomplete in terms of accuracy [31], we assumed results from DETOKS as groundtruth during inspection to shed light on how models are utilizing important aspects in data. In Section 5.4, we compared patterns detected by DETOKS algorithm and templates constructed in our method.

Saliency Map for Interpreting the Predictions
In a perspective of model interpretability, we show which part of EEG recordings models have paid attention to when making a prediction. Saliency Map [37] calculates gradient value, or saliency value, of model’s output with respect to each data point. Intuitively, data points with high saliency values indicate significant parts exploited during prediction, since model output is more severely affected by changes in those regions compared to other regions. As proposed in the original work, to reveal how models utilized features in input recordings, we calculated the derivative of models output prediction with respect to input recording: \( \frac{\partial f(x)}{\partial x} \), where \( f(x) \) is a prediction of model.

4.5 Implementation Details

Datasets were divided into train, validation and test data with a ratio of 80%, 10% and 10% on subject basis, which means that recording segments from a subject cannot be contained in both of training and test data. We chose subject-based division scheme to avoid information leak between train and test data. All the recordings were annotated and classified for each of 30 second segment. EEG signals were then re-sampled to 100 Hz before being fed into the model. Reported performances in result section were produced from models with the best Macro-F1 score on validation dataset. Models were implemented in Python 3.6.7 and Pytorch 1.4.0 [32]. We executed experiments on servers equipped with NVIDIA GeForce RTX 2080 Ti GPUs and Intel Xeon(R) Gold 5118 2.30GHz CPUs.

5 RESULTS

In this section, we refer (1) models trained with the proposed work as Template and (2) models from the original work of target neural networks as Baseline. Results on classification performances and confidence calibration are provided. Afterwards, we estimated robustness in several aspects. Lastly, we visually investigated and interpreted learned representations.

5.1 Classification Performances

Table 3 lists the classification performances measured from large datasets containing 100 recordings or more. Throughout several architectures, models trained with the proposed method achieved better scores compared to baseline architectures in terms of Macro-F1 scores and accuracies. For Sors, although our model performed better on EDF-2018 and CAP dataset, differences in scores are not significant compared to other architectures. In ISRUC, the baseline model was even better compared to ours. We believe such results are due to the small filter size employed in the first convolutional layer of Sors. Since the first convolutional layer has filter length of 7 in Sors, it might be harder for the proposed method to extract template patterns with such short filters. Thus, models might fail in fully utilizing the advantages of template pre-training.

Additional experiments were conducted on small datasets. Macro-F1 scores measured from datasets with fewer than 50 subjects are provided in Table 4. Models trained with the proposed method showed larger improvements in classification performances compared to results reported in Table 3. For example, in ISRUC Group3, Macro-F1 scores for LSTM model are improved by 15% on average by adopting the proposed pre-training, whereas Macro-F1 score increased only by 2% in ISRUC Group1. In Section 5.3, we thoroughly inspected the robustness of the proposed method to dataset size.
Table 4: Classification Performances on Small Datasets

| Models          | Methods | EDF-2013       | ISRUC Group3 | SVUH-UCD       |
|-----------------|---------|----------------|--------------|----------------|
|                 |         | Macro-F1       | Accuracy     | Macro-F1       | Accuracy     |
| Supratak [39]   | Template | 0.78152 ± 0.051 | 0.82728 ± 0.052 | 0.76321 ± 0.046 | 0.76393 ± 0.047 | 0.59814 ± 0.102 | 0.61415 ± 0.114 |
|                 | Baseline | 0.77524 ± 0.047 | 0.81704 ± 0.053 | 0.63802 ± 0.087 | 0.67448 ± 0.090 | 0.57672 ± 0.084 | 0.60235 ± 0.086 |
| Supratak [39]   | Template | 0.72981 ± 0.054 | 0.79175 ± 0.057 | 0.68174 ± 0.051 | 0.73134 ± 0.050 | 0.55026 ± 0.051 | 0.55946 ± 0.056 |
|                 | Baseline | 0.72513 ± 0.049 | 0.79128 ± 0.053 | 0.65717 ± 0.051 | 0.69898 ± 0.070 | 0.52884 ± 0.073 | 0.55744 ± 0.082 |
| Sors [38]       | Template | 0.74566 ± 0.052 | 0.81823 ± 0.049 | 0.62768 ± 0.053 | 0.69159 ± 0.055 | 0.45682 ± 0.051 | 0.49390 ± 0.069 |
|                 | Baseline | 0.74265 ± 0.050 | 0.81902 ± 0.047 | 0.62057 ± 0.051 | 0.68300 ± 0.058 | 0.44484 ± 0.052 | 0.47606 ± 0.061 |
| Dai [6]         | Template | 0.76623 ± 0.056 | 0.81957 ± 0.059 | 0.65811 ± 0.055 | 0.71086 ± 0.052 | 0.58210 ± 0.074 | 0.60416 ± 0.094 |
| 5-layer CNN     | Baseline | 0.74841 ± 0.060 | 0.80078 ± 0.060 | 0.67422 ± 0.056 | 0.72598 ± 0.048 | 0.57016 ± 0.073 | 0.58874 ± 0.088 |
| Dai [6]         | Template | 0.75892 ± 0.064 | 0.81894 ± 0.062 | 0.68474 ± 0.063 | 0.73363 ± 0.056 | 0.58110 ± 0.086 | 0.59897 ± 0.098 |
| 11-layer CNN    | Baseline | 0.74894 ± 0.056 | 0.80500 ± 0.056 | 0.67806 ± 0.050 | 0.72850 ± 0.054 | 0.57033 ± 0.078 | 0.58527 ± 0.094 |

Table 5: ECE Scores Calculated for Sleep Datasets

| Dataset         | Supratak | Sors   |
|-----------------|----------|--------|
|                 | Baseline | Template | Baseline | Template |
| EDF-2013        | 14.53    | 10.96   | 14.38    | 14.21    |
| EDF-2018        | 14.45    | 11.09   | 16.28    | 13.91    |
| ISRUC Group1    | 17.95    | 13.26   | 11.49    | 13.52    |
| ISRUC Group3    | 25.71    | 15.32   | 9.68     | 10.50    |
| CAP             | 19.49    | 15.79   | 13.73    | 12.10    |
| SVUH-UCD        | 33.48    | 29.49   | 21.44    | 19.77    |

5.2 Evaluation on Confidence Calibration

We examined the robustness of the proposed method regarding confidences in the output. Similar to previous works [13], maximum values from softmax distribution of model’s outputs were interpreted as confidence scores. Table 5 lists ECE scores calculated from the architecture proposed in Supratak’s work [39]. For models built on recurrent layers, ECE scores on softmax confidence decreased significantly when TRIER was adopted. Reduced errors on confidence estimation indicate that the proposed method helped models to produce calibrated outputs. Such results are in concordance with the findings that pre-training can enhance robustness regarding model uncertainty [14]. For Sors, confidence calibration was better in ISRUC datasets. However, we believe significant improvements on confidence calibration in Supratak architecture are still meaningful, since modeling a temporal relations between sleep stages with recurrent layers is widely adopted in sleep staging tasks [27, 34, 39].

To further evaluate the degree of calibration on each confidence interval, we generated reliability diagram in Figure 1. It can be observed that confidences from the proposed method are overall more close to ideal calibration levels. This indicates that our pre-training scheme effectively relieved the over-confident tendency of neural networks [12].

5.3 Evaluation on Robustness

Performances with Varying the Dataset Size

To show that the proposed method works out robustly regardless of dataset size, we evaluated performances of models with varying the size of train dataset. Experiments were conducted on datasets with more than 100 subjects: EDF-2018, ISRUC Group1 and CAP dataset. Results are listed in Table 6 and Table 7. The proposed method consistently shows better performances throughout various dataset sizes. Similarly, ECE scores are significantly lower. These results indicate that models trained from the proposed strategy are robust under different dataset sizes. This might be owing to the ability...
that our method can obtain more generalizable features, which are based on template patterns.

**Robustness to Noise**

Since bio-physical signals are easily contaminated by noises [10], it is worth investigating the robustness of models against noises. Among various forms of noises, we experimented using gaussian noises, which were extensively considered in researches on model robustness [8, 15]. Noise scales were selected with respect to standard deviations of EEG recordings. Figure 2 demonstrates the performances of models in terms of classification accuracy and confidence calibration. Models trained with the proposed method show better performances with little decrease in Macro-F1 scores until the noise scale reaches 0.3. Even though Macro-F1 scores do not show significant differences in EDF dataset, performances are consistently higher than the baseline models. Figure 2b shows ECE scores on various noise levels. As shown in the figure, ECE scores from the proposed method begin to increase at higher noise scales, whereas confidence calibration deteriorates right after the noises are injected. These results demonstrate that owing to template patterns constructed during pre-training, our models succeed in reliably predicting sleep stages from contaminated EEG recordings with better classification performances.

5.4 Investigation and Interpretation on Model Analysis on Learned Filters

To demonstrate that the proposed method effectively learned template patterns, filters in the first convolutional layer of the employed architectures were extensively inspected. We investigated whether convolutional filters correspond to well-known patterns in sleep staging. We applied DETOKS algorithm [31] to the filters. As shown in Figure 3, only the filters from the proposed method correctly contain significant patterns like Sleep Spindles and K-complexes. Even though some false positives were detected as K-complexes in Figure 3, only the filters from the proposed method correctly contain significant patterns like Sleep Spindles and K-complexes.

![Figure 2: Performances Measured on Gaussian Noises](image)

![Figure 3: Analysis on Learned Filters](image)
Figure 3: Overall Shapes of Convolutional Filters from Supratak [39]. Filters from the proposed method and Baseline are drawn. Regions corresponding to Sleep Spindles and K-complexes detected by DETOKS are highlighted by different colors. On spectrogram, filters that are dedicated to certain frequency bands are marked with boundary colors.

Figure 4: Interpretations on Supratak [39] When Predicting N2 Stage. In sub-figures, EEG recordings from N2 stage are plotted. Regions corresponding to ten highest saliency values were highlighted. Highlights in the first row are drawn with the saliency map from the proposed method and the second row corresponds to that of Baseline. In addition, K-complex and Sleep Spindles captured by DETOKS are marked with boundary boxes.

marked their spectrograms with boundaries of different colors for each frequency band. As shown in Figure 3, models from the proposed method successfully learned templates of theta waves, delta waves and fast waves.

**Interpretation Based-on Template Patterns**
As Sleep Stage N2 contains characteristic EEG patterns, Sleep Spindles and K-complexes, that are only observed during the stage with distinctive shapes, we mainly interpreted EEG segments corresponding to N2 stage. We conducted interpretation experiments focusing on whether models effectively located Sleep Spindles and K-complexes in recordings. We believe such interpretation experiments are meaningful, since experts also utilize presence of these patterns when annotating sleep stages for EEG data [26, 40]. In Figure 4, we draw raw EEG signals with highlights on sub-series which have high saliency values. Furthermore, we annotated regions corresponding to Sleep Spindles and K-complexes that were detected by DETOKS [31]. Comparing the saliency maps from the proposed method and the baseline architecture, both models effectively located Sleep Spindles in data, which can be characterized by 12-15 Hz bursts. However, the baseline model did not capture features corresponding to K-complexes, whereas models trained by the proposed method successfully capture K-complexes. We believe such differences are owing to the template filters that resemble K-complexes inspected in the previous section. These results indicate that our method successfully reconstructed template patterns that
are referenced by trained experts and models trained with the proposed method can deal with well-known patterns corresponding to each sleep stage.

**Visualizing Robustness to Noises**

To investigate why our method is more robust to gaussian noises, we further visualized the changes in saliency values when noises are injected to original data. We believe we can gather understandings on how models respond to noises by inspecting such changes. For this reason, we selected EEG segments that were correctly predicted from both of the proposed method and the baseline model, but misclassified by the baseline model when noises were injected. In Figure 5, as well as raw recordings, we draw four saliency maps from the proposed method and the baseline architecture with respect to two data: (1) original and (2) noise-contaminated recordings. As shown in the figure, saliency values from the proposed method were relatively consistent under noises. For example, in Figure 5a, we can see that high-saliency regions in the front part \((t < 1000)\) and middle part \((1300 < t < 1800)\) remain unchanged for the proposed method even with noises. In contrast, for the baseline model, saliency values are largely affected by noises with high-saliency regions in the middle part \((1300 < t < 1800)\) mostly decrease. Likewise, in Figure 5c, high-saliency regions in middle part \((1000 < t < 1300)\) shift toward front in the baseline model. For Figure 5d, saliency values between 1500 and 1800 have been nearly zeroed out in the baseline architecture. In summary, it can be observed that overall saliency maps are relatively unchanged in the proposed method whereas there are significant changes in saliency values for the baseline model. These observations suggest that feature space remains stable under noises when representations are built on template patterns.

### 6 CONCLUSION

In this work, we propose a pre-training strategy for neural networks on sleep staging that enables models to utilize template patterns in EEG recordings. We propose a novel neural network, where cosine similarity is introduced as basic operator in order to effectively extract template signals based on significant waveform shapes. Although use of cosine similarity in neural networks has already been introduced in the previous work [24], as long as we know, this is the first work to utilize cosine similarity to better extract meaningful representations considering the irregular characteristics of data. Extracted patterns were used in initializing the first convolutional layer of target models. With extensive experiments, we revealed that the proposed method works generally well on sleep staging tasks. As well as classification accuracy, overall capacity of the network, including confidence calibration, noise robustness and robustness against small dataset size, was improved by significant degree. Thorough inspection on learned representations and model interpretations demonstrated that models trained with the proposed method were able to utilize important aspects in data that were not handled by the baseline architecture. In summary, template-based representations constructed with the proposed method successfully augmented models in sleep staging tasks in several aspects.

**REFERENCES**

[1] Khald Al Aboalayon, Helen T Ochagabir, and Miad Faezipour. 2014. Efficient sleep stage classification based on EEG signals. In IEEE Long Island Systems, Applications and Technology (LISAT) Conference 2014 IEEE, 1–6. https://doi.org/10.1109/LISAT.2014.6845193

[2] Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. 2015. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. Nature biotechnology 33, 8 (2015), 831–838. https://doi.org/10.1038/nbt.3300

[3] Fernando Andreotti, Huy Phan, Navin Cooray, Christine Lo, Michele TM Hu, and Maarten De Vos. 2018. Multichannel sleep stage classification and transfer learning using convolutional neural networks. In 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, 171–174.
