Towards More Accurate Automatic Sleep Staging via Deep Transfer Learning

Huy Phan*, Oliver Y. Chén, Philipp Koch, Zongqing Lu, Ian McLoughlin, Alfred Mertins,
and Maarten De Vos

Abstract— *Background:* Despite recent significant progress in the development of automatic sleep staging methods, building a good model still remains a big challenge for sleep studies with a small cohort due to the *data-variability* and *data-inefficiency* issues. This work presents a deep transfer learning approach to overcome these issues and enable transferring knowledge from a large dataset to a small cohort for automatic sleep staging. *Methods:* We start from a generic end-to-end deep learning framework for sequence-to-sequence sleep staging and derive two networks as the means for transfer learning. The networks are first trained in the source domain (i.e. the large database). The pretrained networks are then finetuned in the target domain (i.e. the small cohort) to complete knowledge transfer. We employ the Montreal Archive of Sleep Studies (MASS) database consisting of 200 subjects as the source domain and study deep transfer learning on three different target domains: the Sleep Cassette subset and the Sleep Telemetry subset of the Sleep-EDF Expanded database, and the Surrey-cEEGrid database. The target domains are purposely adopted to cover different degrees of data mismatch to the source domains. *Results:* Our experimental results show significant performance improvement on automatic sleep staging on the target domains achieved with the proposed deep transfer learning approach. *Conclusions:* These results suggest the efficacy of the proposed approach in addressing the above-mentioned data-variability and data-inefficiency issues. *Significance:* As a consequence, it would enable one to improve the quality of automatic sleep staging models when the amount of data is relatively small. [1]

Index Terms— Automatic sleep staging, sequence-to-sequence, deep learning, transfer learning.

I. INTRODUCTION

Sleep scoring [1], [2] aims to determine sleep stages from polysommography (PSG) recordings. In clinical environments, this task has been mainly performed manually by clinicians following developed guidelines [1], [2]. Since the manual scoring is time-consuming, costly, and prone to human errors, automating the scoring process has been a long-lasting focus in the sleep research community [3]–[10]. Automatic sleep scoring is particularly important in home-based sleep monitoring [11]–[14]. Recent years have seen a new generation of mobile electroencephalography (EEG) devices that provide a cost-effective solution to screen a wide population for epidemiological studies and to monitor specific populations at risk of sleep disorders.

Deep learning has been successfully applied to numerous domains and has received much attention from the sleep research community. Past work has looked at various deep network architectures, such as deep neural networks (DNNs) [15], convolutional neural networks (CNNs) [7], [8], [10], [16]–[24], and recurrent neural networks (RNNs) [25]–[28], and novel ways to carry out sleep staging like sequence-to-sequence classification scheme [4], [6]. Reviews of the most recent progress on deep learning for automatic sleep staging can be found in [29]–[31]. However, considerably less attention has been paid to make sleep staging models more robust to the challenges of sleep data variability and to make them data-efficient (i.e. using less data). Despite the fact that the performance of machine's sleep staging has been on par with manual scoring by sleep experts [4], [6], [8], [20]–[23], we have not seen it widely adopted clinically. This is arguably due to two major technical drawbacks of the sleep staging models: *data variability* and *data inefficiency*.

Data variability: PSG signals recorded in a particular recording setup are characterized by a number of parameters such as sensors' frequency response and output level, and signal processing applied to the raw signal. These factors contribute to the *transfer function* of the recording device and affect how the physiological signals are converted into digital PSG

H. Phan is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, UK. I. McLouglin is with Singapore Institute of Technology, Singapore 138683. O. Y. Chén is with the Institute of Biomedical Engineering, University of Oxford, Oxford OX3 7DQ, UK. Z. Lu is with the Department of Computer Science, Peking University, Beijing, 100080, China. P. Koch and A. Mertins are with the Institute for Signal Processing, University of Lübeck, Lübeck 23562, Germany. M. De Vos is with the Department of Electrical Engineering and the Department of Development and Regeneration, KU Leuven, 3001 Leuven, Belgium.

*Corresponding author: h.phan@qmul.ac.uk

[1]The source code and the pretrained models are published at http://github.com/pquochuy/sleep_transfer_learning.

TABLE I: Out-domain performance of the single-channel SeqSleepNet+ trained on MASS database in comparison to its in-domain performance.

Database	MASS	Sleep-EDF-SC	Sleep-EDF-ST	Surrey-cEEGrid
Input	C4-A1	Fpz-Cz	Fpz-Cz	cEEGrid
Accuracy	84.5 (in-domain)	81.2 (out-of-domain)	80.5 (out-of-domain)	10.6 (out-of-domain)
Mismatch	-	slight	slight	severe

output. As a result, sleep data recorded in different setups may have different transfer functions due to the variations in their underlying hardware and software processing pipelines. Furthermore, discrepancies in channel layouts [8] and cohort characteristics [32] are also likely in different sleep studies. From the viewpoint of machine learning models, these variations and discrepancies lead to *domain shift* or mismatch between sleep data sources. Data mismatch across different acquisition conditions are computationally significant, degrading the accuracy of sleep staging models on unseen data with a novel recording condition. Therefore, if a sleep staging model is deployed on an unseen sleep data whose properties differ from the data used for training the model, the data mismatch can result in poor inference performance. As evidenced in our experiments (cf. Section V-C) and shown in Table I, the single-channel SeqSleepNet [4] model trained on the MASS database suffers from an accuracy drop when it is evaluated *out of domain* (i.e. being tested on other three databases Sleep-EDF-SC, Sleep-EDF-ST, and Surrey-cEEGrid) relative to the one obtained with *in-domain* evaluation (i.e. via cross-validation on the MASS database itself). The performance loss depends on the level of data mismatch.

Data inefficiency: Existing deep-learning based sleep staging models cannot escape from the curse of data inefficiency of the deep learning paradigm. That is, training a deep neural network generally requires a large amount of data. In fact, expert-level performance on automatic sleep staging is only obtainable with these models when the training cohort is large, i.e. hundreds or thousands of subjects [4], [8]. The networks' performances decline significantly when they are trained with a small cohort (e.g. ten or twenty subjects [6], [33]). Unfortunately, in practice, many sleep studies only have access to a small cohort, in the order of a few dozens of subjects [11], [32], [34]–[37]. Thus, the small data in these studies hinder deep learning models to perform well.

An easy and obvious solution for the above-mentioned obstacles is to collect training data from all types of recording setups (e.g. recording devices, channel layouts, and preprocessing softwares) that will be foreseeably encountered in the deployment phase. However, this is an expensive, time-consuming, and even infeasible solution. First, most of large sleep databases are proprietary, making those inaccessible for research purposes. Second, even if they are available, a huge effort would be required to score these data manually. Third, novel setups will likely emerge when one studies a particular sleep disorder [32], [37] or when one explores the feasibility of a new monitoring device [11].

In this work, we present a practical solution based on transfer learning to tackle these obstacles, to build more accurate sleep staging models when the available data is small, and to recover the performance of the models otherwise lost due to data variability. We leverage a reasonably large sleep database, which is publicly available, and use a sleep-staging deep neural network as a device to transfer knowledge from this database to improve sleep staging performance on another small cohort with a different recording setup. More specifically, the network is firstly trained with the large database (the source domain) and subsequently finetuned with the small

cohort (the target domain) to complete transfer learning. In this context, *finetuning* means a part or the entire of the pretrained network is further trained with the target domain data. The main contributions of this work include:

- A new perspective of looking at data variability and data inefficiency in the automatic sleep staging problem, and developing a deep transfer learning approach to overcome sleep data mismatch and enable knowledge transfer to improve sleep-staging performance on small cohorts. In-depth investigation into the influence of the number of subjects on the transfer learning performance was also conducted.
- The generalization of a sequence-to-sequence sleep staging framework from which two state-of-the-art models *SeqSleepNet+* and *DeepSleepNet+* are developed and used in the study.
- A systematic study highlighting different target domains with varying data-mismatch degrees to the source domain, different transfer learning scenarios (i.e. single-channel and multi-channel input), different finetuning strategies, and different state-of-the-art sleep staging models. Our transfer learning approach outperforms all the tested baselines and existing works in solving the automatic sleep staging on the target sleep databases.

This work extends our preliminary work in [33] in several aspects. First, we study transfer learning with a wider spectrum of channel combinations for the networks' input rather than a single channel. Second, the studies in [33] employed SeqSleepNet [4] as the transfer learning device, here the studies are carried out on two different networks inherited from SeqSleepNet [4] and DeepSleepNet [6]. These two state-of-the-art networks are diverging in their architectures [38]; therefore, it is important to examine if these dissimilarities give rise to any difference in their performance and to explain their behaviors in transfer learning. Third, the work in [33] only studied deep transfer learning on the Sleep-EDF-SC as the target domain. Here, we cover multiple target domains with varying degrees of channel mismatch. Fourth, we study in-depth the influence of the number of target subjects on the transfer learning's performance.

II. MATERIALS

A. Source Domain

We adopted the public Montreal Archive of Sleep Studies (MASS) database [39] as the source domain in this study as it is sufficiently large.

MASS: This database was pooled from different hospital-based sleep laboratories, consisting of whole-night recordings from 200 subjects (97 males and 103 females) aged between 18 and 76 years. Manual annotation was accomplished by sleep experts according to the AASM standard [1] (SS1 and SS3 subsets) or the R&K standard [2] (SS2, SS4, and SS5 subsets). As in [4], [5], we converted different annotations into five sleep stages {W, N1, N2, N3, and REM} and expanded 20-second epochs into 30-second ones by including 5-second segments before and after each epoch. We used the C4-A1 EEG, ROC-LOC EOG, and CHIN1-CHIN2 EMG in our experiments.
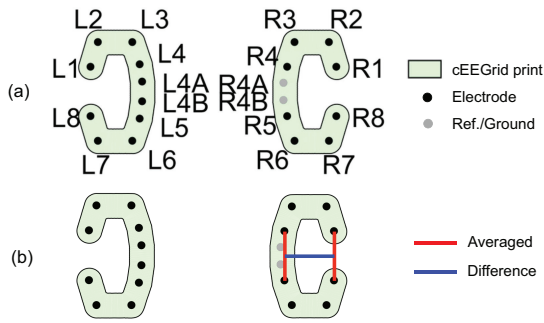
Fig. 1: Illustration of (a) the cEEGrid electrode array which was used to record the Surrey-cEEGrid database [11], [40] and (b) the FB(R) ("front versus back" for the right ear) derivation [11] used in our experiments.

B. Target Domains

Three different sleep databases are used as the target domains. These adopted cohorts have diverging health conditions, i.e. healthy (Sleep-EDF-SC) vs. mild sleep difficulty (Sleep-EDF-ST) [34], [35], and channel characteristics (i.e. traditional PSG recording (Sleep-EDF-SC and Sleep-EDF-ST) vs. wearable around-the-ear EEG recordings (Surrey-cEEGrid) [11], [40]).

Sleep-EDF-SC: This is the Sleep Cassette (SC) subset of the Sleep-EDF Expanded dataset [34], [35], consisting of 20 subjects aged 25-34. Two subsequent day-night PSG recordings were collected for each subject, except for subject 13 who has only one-night data. Each 30-second PSG epoch was manually labelled into one of eight categories {W, N1, N2, N3, N4, REM, MOVEMENT, UNKNOWN} by sleep experts according to the R&K standard [2]. Similar to previous works [5]–[7], [17], [25], [41], N3 and N4 stages were merged into a single stage N3 and MOVEMENT and UNKNOWN categories were excluded. Since full EMG recordings are not available, we only adopted the Fpz-Cz EEG and ROC-LOC EOG (i.e. the EOG horizontal) channels in this study. As this database has been used differently in literature, it should be stressed that only the in-bed parts (from *lights off* time to *lights on* time) of the recordings were used as recommended in [5], [7], [17], [25], [41]–[43].

Sleep-EDF-ST: This is the Sleep Telemetry (ST) subset of the Sleep-EDF Expanded dataset [34], [35] which was collected for studying the temazepam effects on sleep. The subset consists of 22 Caucasian subjects (7 males and 15 females) aged 18-79 with mild difficulty falling asleep. Although the PSG signals were recorded for two nights, one after temazepam intake and one after placebo intake, only the placebo nights are available. Manual annotation was done similar to the Sleep-EDF-SC subset. Beside Fpz-Cz EEG and ROC-LOC EOG, the submental EMG channel is available and additionally adopted. Similar to the the Sleep-EDF-SC subset, only the in-bed parts of the recordings were used.

Surrey-cEEGrid: This database [11], [40] was recorded at the University of Surrey using the cEEGrid array [44], [45], a novel lightweight flex-printed electrode strip that fits neatly behind the ear, as illustrated in Figure 1 (a). Twenty participants, aged 34.9 ± 13.8 years, had their overnight

TABLE II: Summary of the employed sleep databases.

	Num. of subjects	EEG	EOG	EMG	Data mismatch
MASS	200	C4-A1	ROC-LOC	CHIN1-CHIN2	-
Sleep-EDF-SC	20	Fpz-Cz	ROC-LOC	-	slight
Sleep-EDF-ST	22	Fpz-Cz	ROC-LOC	Submental	slight
Surrey-cEEGrid	12	cEEGrid	ROC-A2	CHIN1-CHIN3	severe

(about 12 hours) cEEGrid data collected. The PSGs were also recorded in parallel and manual annotation based on the PSG was used as reference for the cEEGrid data [11]. Besides two recordings lost due to human error, six recordings were discarded because of excessive artifacts and missing data. A cohort of 12 participants was retained. From the cEEGrid data, the FB(R) ("front versus back" for the right ear, see Figure 1 (b)) EEG derivation, which was the best derivation [11], was obtained and used. We also simulated the two- and three-channel settings by adding the ROC-A2 EOG and CHIN1-CHIN3 channels from the PSG data to the cEEGrid data. Although there exist other EOG and EMG channels, the ROC-A2 EOG and CHIN1-CHIN3 channels were deliberately selected to be different from those of the source domain to maintain the severity of data mismatch.

The employed databases and the adopted signals are summarized in Table II. All the signals were downsampled to 100 Hz. The databases were chosen to have the data mismatch between the target domains and the source domain varying from slight level due to the difference in PSG signals used (i.e. Sleep-EDF-SC and Sleep-EDF-ST) to severe level due to completely new electrode placement (i.e. Surrey-cEEGrid).

III. THE GENERIC DEEP LEARNING FRAMEWORK FOR SEQUENCE-TO-SEQUENCE SLEEP STAGING

The advent of deep learning has made astonishing progress in automatic sleep staging. First, deep networks are powerful in learning features which outperform and displace traditional handcrafted features. Second, they enable us to achieve automatic sleep stage classification in ways that are impossible for conventional machine-learning algorithms. The sequence-to-sequence sleep staging scheme [4] was recently proposed to offer the ability of modelling long-term temporal dependency of sleep data epochs in a deep learning model. Intuitively, a sequence-to-sequence model processes a sequence of multiple consecutive epochs simultaneously and classifies them at once into a sequence of corresponding sleep stages. Here, we frame this scheme into a generic deep learning framework for sequence-to-sequence sleep staging. This framework also sets a potential benchmark to design new models in future work. It is worth noting beforehand that a detail explanation of the network layers and machine learning concepts encountered in the following sections, such as an RNN or a CNN, can be found in [46].

A. The framework

Formally, given the input sequence of L consecutive epochs denoted as $(\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_L)$, the sequence-to-sequence sleep staging problem [4] is formulated to maximize the conditional probability $p(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L \,|\, \mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_L)$ where
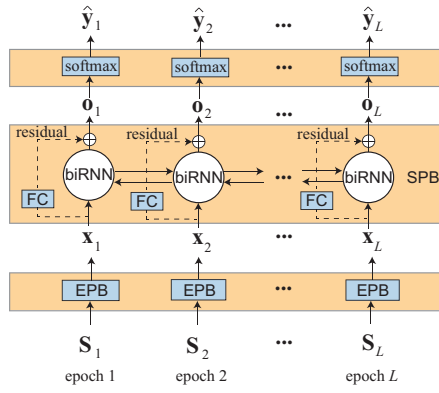
Fig. 2: The proposed generic deep learning framework for sequence-to-sequence sleep staging which consists of the tied epoch processing block (EPB), the sequence processing block (SPB), and the tied softmax layer.
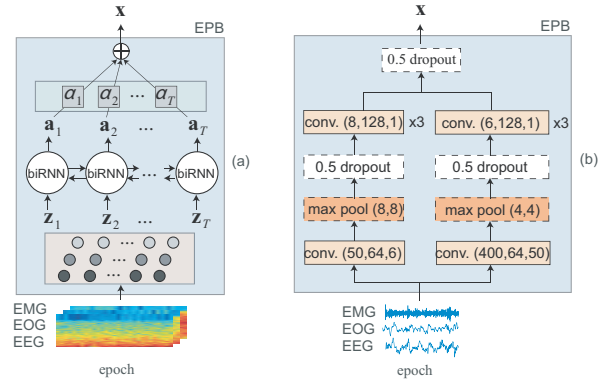

Fig. 3: Illustration of the EPBs of (a) SeqSleepNet+ and (b) DeepSleepNet+. The former relies on an attentional biRNN coupled with filterbank layers. The latter is a two-branch deep CNN.

$(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L)$ represents the sequence of corresponding L one-hot encoding vectors of the ground-truth output labels.

The proposed framework are divided into three components, an epoch processing block (EPB), a sequence processing block (SPB), and a softmax layer, as illustrated in Fig. 2.

EPB: Each epoch in the input sequence is presented to the network in some forms of representation (e.g. raw signals [6] or time-frequency features [4]) and can be single-channel (e.g. EEG or EOG) or multi-channel (e.g. a combination of EEG, EOG, and EMG). The EPB plays the role of an epoch-wise feature learner and extractor. The EPB is common for the PSG epochs in the input sequence and is a sub-network that is trained jointly with other components in an end-to-end manner [4]. Via the EPB, an input epoch $\mathbf{S}_l$, $1 \leq l \leq L$, is transformed into an epoch-wise feature vector $\mathbf{x}_l$.

SPB: The SPB consists of a bidirectional recurrent layer (biRNN) that encodes the sequence of the induced epoch-wise feature vectors $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L)$ into the sequence of output vectors $(\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_L)$. An RNN is a type of deep neural networks that processes an input sequence one element at a time and retain information of all the past elements of the sequence in its hidden state vector [47]. A biRNN, on the other hand, consists of two RNN layers of opposite directions to the same input sequence [48]. More specifically, the forward and backward recurrent layers of the biRNN iterate over the sequence $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L)$ in opposite directions and compute their forward and backward sequences of hidden state vectors $\mathbf{H}^{\mathrm{f}} = (\mathbf{h}_1^{\mathrm{f}}, \mathbf{h}_2^{\mathrm{f}}, \ldots, \mathbf{h}_L^{\mathrm{f}})$ and $\mathbf{H}^{\mathrm{b}} = (\mathbf{h}_1^{\mathrm{b}}, \mathbf{h}_2^{\mathrm{b}}, \ldots, \mathbf{h}_L^{\mathrm{b}})$, respectively, where

$$\mathbf{h}_l^{\mathrm{f}} = \mathcal{H}(\mathbf{x}_l, \mathbf{h}_{l-1}^{\mathrm{f}}), \tag{1}$$

$$\mathbf{h}_l^{\mathrm{b}} = \mathcal{H}(\mathbf{x}_l, \mathbf{h}_{l+1}^{\mathrm{b}}), \ 1 \leq l \leq L. \tag{2}$$

In (1) and (2), $\mathcal{H}$ denotes the hidden layer function of the biRNN and can be realized by either Long Short-Term Memory (LSTM) [49] or Gated Recurrent Unit (GRU) [50], two most popular RNN variants. The sequence of output vectors $(\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_L)$ is then computed:

$$\mathbf{o}_l = \mathbf{W}_{ho}[\mathbf{h}_l^{\mathrm{b}} \oplus \mathbf{h}_l^{\mathrm{f}}] + \mathbf{b}_o, \ 1 \leq l \leq L, \tag{3}$$

where $\oplus$ represents vector concatenation. In (3), $\mathbf{W}_{ho}$ denotes a learnable weight matrix and $\mathbf{b}_o$ denotes a learnable bias. The (long-term) dependency of the input epochs are expected to be modelled by the biRNN layer and the output vectors $\mathbf{o}_l$, $1 \leq l \leq L$ are expected to encode sequence-level information. A residual connection can be optionally used to integrate epoch-wise features $\mathbf{x}_l$ and sequence-wise features $\mathbf{o}_l$ and, hence, enables the network to explore their combination in the classification stage. The fully-connected layer (FC) of the residual connection is to convert $\mathbf{x}_l$ into another vector having its size compatible to $\mathbf{o}_l$ for a proper residual combination. All the residual connections also share their parameters.

Softmax: The classification is carried out by the shared softmax layer to yield the output sequence of sleep stage probabilities $(\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots, \hat{\mathbf{y}}_L)$ from the sequence of output vectors $(\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_L)$. Different from SeqSleepNet in [4] and DeepSleepNet in [6], we use a common softmax layer for classification at all indices $1, 2, \ldots, L$ to reduce the number of network parameters rather than one separate softmax layer at each of the indices. A network that adheres to this framework can be trained to minimize the sequence classification loss over N training sequences in the training data:

$$E(\boldsymbol{\theta}) = -\frac{1}{L} \sum_{n=1}^{N} \sum_{l=1}^{L} \mathbf{y}_l \log\left(\hat{\mathbf{y}}_l\left(\boldsymbol{\theta}\right)\right) + \frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2. \tag{4}$$

Here, $\boldsymbol{\theta}$ represents the network parameters and λ denotes the hyper-parameter that trades off the error terms and the ℓ_2-norm regularization term.

B. The derived networks

From the framework presented in Section III-A, we develop two networks as the base models for transfer learning:

SeqSleepNet+: This network is similar to SeqSleepNet presented in [4], except that a common softmax layer is used at all indices of the input sequence. Hence, SeqSleepNet+ is more compact than SeqSleepNet [4]. The network receives the log-scale time-frequency representation [4] as input. The time-frequency image is normalized to zero-mean and unit standard deviation. In case of multi-channel, the channel-wise image features are stacked as a multi-channel image. The network's

EPB is realized by *filterbank* layers [4], [17], one for each input image channel for preprocessing purpose, followed by an attentional biRNN as illustrated in Figure 3(a). Note that this EBP's biRNN should not be confused with the SBP's biRNN in Fig. 2. Both the EPB's biRNN and the SPB's biRNN of the network are implemented by a GRU cell [50] with recurrent batch normalization [51]. There is no residual connection (cf. Figure 2) in the SPB of this network.

DeepSleepNet+: This network is inherited from DeepSleep-Net [6] and its end-to-end variant [4], except for the common softmax used at all indices of the input sequence. The network receives raw signals as input. When the input are composed of multiple signals, the raw signal are stacked to form a multi-channel input. The network's EPB is composed of two deep CNNs organized in two branches with 4 convolutional layers each as illustrated in Figure 3(b). A CNN is a type of deep neural networks designed to efficiently process data that come in the form of multiple arrays [47], such as one-dimensional signals in this case. A CNN features local connections, shared weights, and pooling to learn translation-invariant features from the input. The convolutional kernels in the two branches are purposely designed to have different sizes so that they can learn features at both fine and coarse temporal resolutions. Each convolutional layer is associated with batch normal-ization [52] and Rectified Linear Units (ReLU) activation [53]. The SPB's biRNN relies on the LSTM cell [49] and is designed to have two bidirectional LSTM layers, one stacked on top of the other. In addition, the SPB makes use of the residual connection.

As the two networks inherits SeqSleepNet's and DeepSleep-Net's architecture's, respectively, they are divergent in their inputs, EPB, and SPB components [38]. Therefore, these dif-ferences suggest discrepant behaviors during transfer learning.

IV. Transfer Learning Scenarios for Automatic Sleep Staging on Small Cohorts

Formally, let $\mathcal{D}_S = \{\mathcal{X}_S, \mathcal{Y}_S\}$ denote the source domain with the feature space $\mathcal{X}_S$ and the label space $\mathcal{Y}_S$. In addition, let $\mathcal{T}_S$ denote the task in the source domain with the source conditional probability distributions $P(\mathbf{y}_S \,|\, \mathbf{x}_S)$, where $\mathbf{x}_S \in \mathcal{X}_S$ and $\mathbf{y}_S \in \mathcal{Y}_S$. Similarly, $\mathcal{D}_T = \{\mathcal{X}_T, \mathcal{Y}_T\}$ denotes the target domain with the feature space $\mathcal{X}_T$ and the label space $\mathcal{Y}_T$. $\mathcal{T}_T$ denotes the task in the target domain with the conditional probability distributions $P(\mathbf{y}_T \,|\, \mathbf{x}_T)$, where $\mathbf{x}_T \in \mathcal{X}_T$ and $\mathbf{y}_T \in \mathcal{Y}_T$, respectively. The objective of transfer learning is to improve learning $P(\mathbf{y}_T \,|\, \mathbf{x}_T)$ with information gained from $\mathcal{D}_S$ and $\mathcal{T}_S$ where $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$ [54]. In our case, $\mathcal{T}_S \equiv \mathcal{T}_T$, as we aim at performing sleep staging with the same set of sleep stages in both the source and target domains. Transfer learning [54] relaxes the hypothesis that the training data must be identically distributed as the test data. Therefore, it is useful to deal with data mismatch and holds promise to leverage the large amount of available data to overcome the problem of having insufficient training data in small cohort studies.

In the present context, a model (e.g. SeqSleepNet+ or DeepSleepNet+) is firstly trained in the source domain and then finetuned in the target domain to complete knowledge
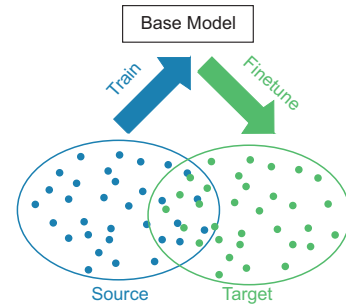

Fig. 4: Transfer learning from a source domain to a target domain. The base model is trained using the source-domain data, and then finetuned on the target-domain data to complete knowledge transfer.

transfer as illustrated in Figure 4. Without loss of generality, the pretraining process is to minimize the loss L_S over the source-domain data, resulting in the model parameter $\boldsymbol{\theta}$:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} = \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}_S} L_S\left(\mathbf{x}, P(\mathbf{y} \,|\, \mathbf{x}), P_{\boldsymbol{\theta}}\left(\mathbf{y} \,|\, \mathbf{x}\right)\right). \quad (5)$$

The pretrained model is considered as a starting point in the target domain. To accomplish transfer learning, a subset of the pretrained network's parameter $\boldsymbol{\theta}' \subseteq \boldsymbol{\theta}$ is finetuned (i.e. further trained) with the target-domain data while the rest $\boldsymbol{\theta} \backslash \boldsymbol{\theta}'$ remains unchanged (i.e. being reused):

$$\underset{\boldsymbol{\theta}' \subseteq \boldsymbol{\theta}}{\operatorname{argmin}} = \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}_T} L_T\left(\mathbf{x}, P(\mathbf{y} \,|\, \mathbf{x}), P_{\boldsymbol{\theta}}\left(\mathbf{y} \,|\, \mathbf{x}\right)\right). \quad (6)$$

When $\boldsymbol{\theta}' = \boldsymbol{\theta}$, the entire pretrained network is finetuned in the target domain. In contrast, when $\boldsymbol{\theta}' = \emptyset$, no finetuning happens and the pretrained network is directly used in the target domain.

In order to study the influence of finetuning different components of a pretrained SeqSleepNet+ and DeepSleepNet+ to the sleep staging performance on the target domains, we examine four *finetuning strategies* corresponding to different component combinations: all, EPB+softmax, SPB+softmax, and softmax. The parameter subsets corresponding to these combinations will be adapted with the target-domain data while the rest remains fixed. The case in which the pretrained network is directly used in the target domain without finetun-ing is considered as a baseline. The finetuning strategies are carried out to study the following *transfer learning scenarios*:

EEG·EOG·EMG↦EEG·EOG·EMG: Apart from brain activities, sleep also involves eye movements and muscular activities at different levels. For instance, Rapid Eye Move-ment (REM) stage usually associates with rapid eye move-ments and high muscular activities are usually seen during the Awake stage. Therefore, EOG and EMG are valuable additional sources, complementing EEG in the automatic sleep staging task [5], [8]–[10], [37]. We study this three-channel EEG·EOG·EMG transfer learning scenario when all EEG, EOG, and EMG are available in a target domain (i.e. in case of Sleep-EDF-ST and Surrey-cEEGrid).

EEG·EOG↦EEG·EOG: This scenario assumes the un-availability of EMG and examines two-channel EEG·EOG transfer learning. Different from the three-channel case, we

are able to study this scenario across all the adopted target domains as they all have full EEG and EOG recordings available.

EEG↦EEG: This scenario explores single-channel EEG transfer learning. Automatic sleep staging with single-channel EEG is prevalent in literature [17], [25], [41], [55]–[57]. Without the augmentation from EOG and EMG, this single-channel setting usually results in a lower performance compared to those of the multi-channel ones; however, it is desirable due to the simple configuration. It is particularly useful for sleep monitoring applications with mobile EEG devices [11], [40].

EOG↦EOG: In general, EOG signals contain rich information from multiple sources, including ocular activity, frontal EEG activity, and EMG from cranial and eye muscles [36]. They are, therefore, promising alternatives for EEG in single-channel sleep staging. In addition, due to the ease of electrode placements, it would be ideal for home-based sleep monitoring applications with wearable devices [11], [40]. Despite their potential, EOG signals have been mainly used as secondary modality in multi-channel sleep staging studies [36], [58]. With this scenario, we aim to exploit standalone EOG and deep transfer learning on this secondary modality to examine whether its performance is comparable to that using the primary EEG in single-channel sleep staging.

EEG↦EOG: As an extension of the EOG↦EOG scenario, this cross-modality transfer learning scenario investigates whether a base model trained on EEG in the source domain can be transferred to EOG in the target domain and if its performance is comparable to the same-domain EOG↦EOG transfer learning scenario. If the answers to these questions are true, instead of modality-specific pretrained models, a single model pretrained solely on EEG can serve as a generic model for single-channel transfer learning regardless the modality of the target domain.

Apart from the data mismatch caused by the differences in recording devices and/or electrode placements in case of the same-modality scenarios (i.e. EEG·EOG·EMG↦EEG·EOG·EMG, EEG·EOG↦EEG·EOG, EEG↦EEG, and EOG↦EOG), heavy data mismatch is expected in case of the cross-modality EEG↦EOG scenario when the base models are trained with EEG data in the source domain is transferred to EOG data in the target domains. On the one hand, with the same-modality scenarios, we aim to show that even when the source domain and the target domains are of the same modalities, transfer learning is still necessary. On the other hand, the cross-modality scenario is to emphasize that transfer learning is efficient in tackling heavy data mismatch to transfer knowledge from the source domain to the target domains.

V. EXPERIMENTS

A. Experimental Setup

SeqSleepNet+ and DeepSleepNet+ were pretrained using the data from the entire 200 subjects of the MASS database (i.e. the source domain) and then finetuned in the target domains. To evaluate the efficiency of transfer learning on sleep staging in the target domains, cross-validation was conducted.

Leave-one-out cross-validation was conducted for Sleep-EDF-SC (20 subjects), and Surrey-cEEGrid (12 subjects) while 11-fold cross-validation was performed for Sleep-EDF-ST (22 subjects) to have an equal number of test subjects (i.e. 2 subjects) in each cross-validation fold. At each iteration of cross-validation, a number of subjects were randomly selected and left out for validation purpose, i.e. for early stopping the finetuning process, (4 for Sleep-EDF-SC and Sleep-EDF-ST and 2 for Surrey-cEEGrid). The performance over all cross-validation folds was then calculated.

B. Network Parameters

Both SeqSleepNet+ and DeepSleepNet+ were implemented using *Tensorflow* [59]. The networks were parametrized similar to SeqSleepNet and DeepSleepNet in our previous work [4]. We experimented with the input sequence length $L = 20$ epochs as this value is a reasonable choice for these sequence-to-sequence models [4]. The sequences were sampled from the training recordings with a hop size of one epoch for network training and finetuning. During testing, the test sequences were also shifted by one epoch, resulting in an ensemble of L classification decisions at each epoch of a test recordings. A probabilistic aggregation step similar to [4] was carried out to fuse the decision ensemble into the final decision.

In the source domain, the networks were pretrained with the MASS database for 10 training epochs with a minibatch size of 32 sequences. For transfer learning, the pretrained networks were further finetuned on each target-domain databases for 10 finetuning epochs. The finetuning process was stopped early when no accuracy improvement was seen on the validation subjects for 50 finetuning steps. Both network training and finetuning were performed using *Adam* optimizer, an optimization algorithm proposed in [60] for training deep neural networks. This optimizer leverages the power of adaptive learning rates methods to find individual learning rates for each parameter of the network. The initial learning rate of Adam optimizer was set to 10^{-4}.

C. Experimental Results

1) Performance on the source domain: It is first worth assessing SeqSleepNet+ and DeepSleepNet+ on the source domain to see how well they perform on a large number of subjects across the input spectrum. To this end, we conducted 10-fold cross-validation on the source domain. At each iteration, 180 subjects were used for training, 10 for validation, and 10 for testing. The results of the cross-validation folds were finally pooled to calculate the overall metrics, including accuracy, macro F1-score, and Cohen's kappa (κ). The obtained performance with different inputs are shown in Table

TABLE III: Sleep staging performance on the source domain (i.e. the MASS database).

Input	SeqSleepNet			DeepSleepNet		
	Acc.	MF1	κ	Acc.	MF1	κ
EEG·EOG·EMG	87.0	83.3	0.815	86.5	82.4	0.807
EEG·EOG	86.5	82.4	0.808	85.9	81.6	0.799
EEG	84.5	79.8	0.778	84.3	79.7	0.777
EOG	83.9	79.1	0.769	83.7	78.9	0.767

Fig. 5: Performance patterns obtained by finetuning the pretrained SeqSleepNet+ with different finetuning strategies in comparison with those of the SeqSleepNet+ scratch model. (a) EEG·EOG·EMG↦EEG·EOG·EMG, (b) EEG·EOG↦EEG·EOG, (c) EEG↦EEG, (d) EOG↦EOG, and (e) EEG↦EOG.
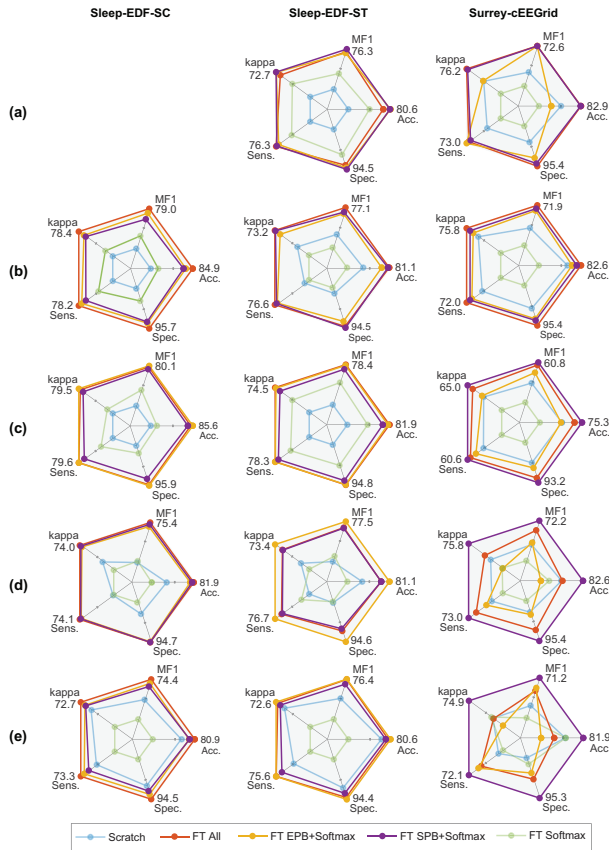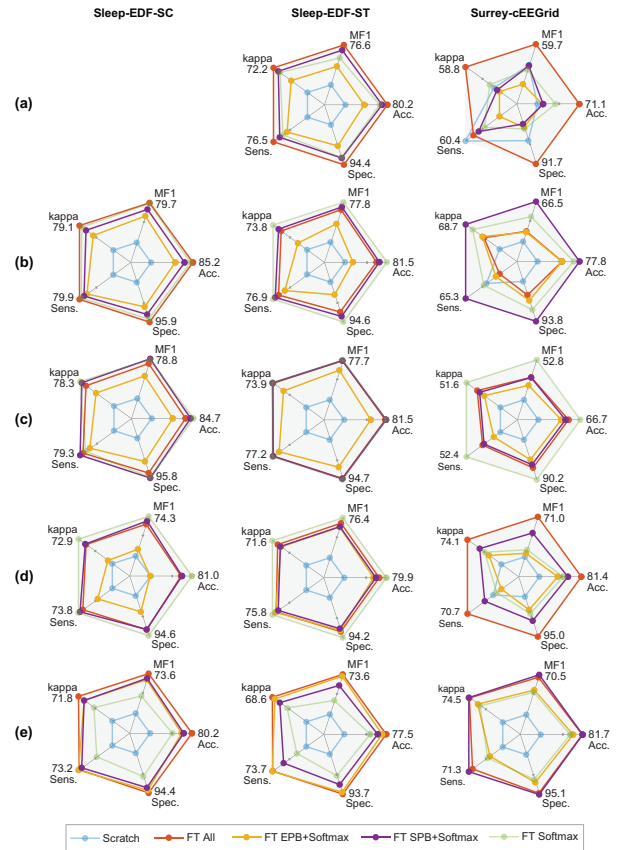
Fig. 6: Performance patterns obtained by finetuning the pretrained DeepSleepNet+ with different finetuning strategies in comparison with those of the DeepSleepNet+ scratch model. (a) EEG·EOG·EMG↦EEG·EOG·EMG, (b) EEG·EOG↦EEG·EOG, (c) EEG↦EEG, (d) EOG↦EOG, and (e) EEG↦EOG.

III. Firstly, the results in the table confirm the benefit of using EOG and EMG to complement EEG in the automatic sleep staging task as their presence lead to performance improvement. Secondly, with the sequence-to-sequence framework, the performance obtained by the secondary EOG is just marginally lower than that of the primary EEG, evidenced by both SeqSleepNet+ and DeepSleepNet+. This suggests that EOG can be used as a standalone modality similar to EEG when a single channel is used.

2) The effect of transfer learning on the target domains:
Figures 5 and 6 give an overall picture on the performance obtained by SeqSleepNet+ and DeepSleepNet+ on the target domains with respect to different finetuning strategies and compared to the model trained from scratch using the target-domain data only. The two networks show noticeably varying patterns on the transfer learning results.

On the one hand, SeqSleepNet+'s results in Figure 5 reveal that, while finetuning the softmax layer alone leads to better performance than that of the scratch model in some cases, it is essential to additionally finetune the feature-learning parts of the network, either the EPB for epoch-level feature learning or the SPB for sequence-level feature learning, or both. This pattern exists across all finetuning cases in the figure. This suggests that the features learned by SeqSleepNet+ in the

source domain are slightly different from those in the target domain. This is reasonable due to the data mismatch between the source and target domains.

On the other hand, DeepSleepNet+'s finetuning results expose diverging patterns as shown in Figure 6. Finetuning the softmax layer alone results in comparable, or even better, performance than other finetuning strategies in some cases (such as the EEG↦EEG scenarios) whereas its results are largely belittled in other cases (such as EEG↦EEG scenarios). This suggests that, when the signals are of the same modality, the features learned from the source domain's raw signals persist in the target domain and only their combinations need to be adapted in the target domains. In contrast, when the signals are from different modalities, additional finetuning the feature learning parts (i.e. the EPB or the SPB or both) is necessary. It, however, should be emphasized that persistence of the learned features across the source and target domains does not necessarily mean good generalization as DeepSleepNet+'s finetuning results are inferior to those of its counterpart, SeqSleepNet+ (see Table IV).

Despite their different behaviors in finetuning, both SeqSleepNet+ and DeepSleepNet+ meet the transfer learning's expectation. Compared to the network trained from scratch using the target-domain data only, transfer learning consis-
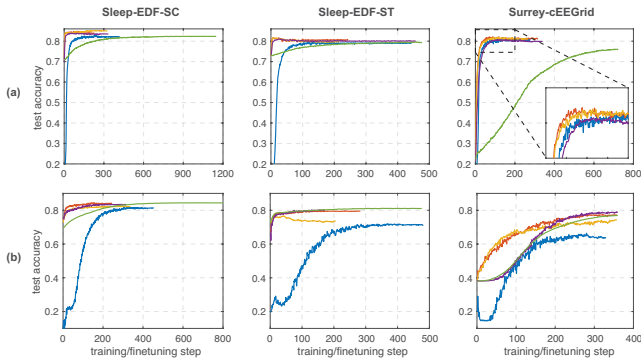
Fig. 7: The test accuracy during the finetuning/training. Each curve was averaged over all the cross-validation folds. (a) SeqSleepNet+ and (b) DeepSleepNet+.

tently results in improvements across different network types, the target domains, and the transfer learning scenarios. The benefits of transfer learning is further evidenced by contrasting the learning curves of the finetuned models and the scratch models. Taking the two-channel EEG·EOG↦EEG·EOG scenario as an example (see Figure 7), the learning curves were recorded on the test data during finetuning and training, respectively. Each learning curve was averaged over all cross-validation folds. As the learning curves' lengths vary across different folds due to early stopping, those with shorter length than the maximum one were padded to the maximum length before averaging. SeqSleepNet+'s learning curves show better generalization and faster convergence of the finetuned models (except the softmax-only finetuning strategy) compared to their scratch opponents. Similar motifs are observed in DeepSleep-Net+'s learning curves; however, the softmax-only finetuning strategy shows a comparable generalization to other strategies (although slower convergence). These findings consolidate the explanation for the finetuning results in Figures 5 and 6.

3) Performance comparison on the target domains: To justify the necessity of transfer learning, in Table IV we compare the finetuning overall performance against those of the scratch models and *direct transfer* (i.e. applying the pretrained models in the target domains without finetuning) across the target domains and the transfer learning scenarios. In addition, the obtained results are also contrasted to those reported in previous works to quantify the efficiency of the proposed transfer learning approach. As the transfer learning results vary depending on the finetuning strategies, for simplicity, out of different finetuning strategies, we retained the SPB+softmax one as the representative for comparison given its consistent finetuning results (see Figures 5 and 6). In practice, the finetuning strategies could be viewed as a hyper-parameter and determined via cross-validation. We should bring to readers' attention a large body of works, such as [8], [21], [23], [24], that yielded an accuracy level on (extremely) large databases similar to that of our proposed systems. However, comparison to these results is not the main focus of this work; furthermore, such a comparison would be incompatible and, hence, does not offer much meaning.

Between SeqSleepNet+ and DeepSleepNet+, the former outperforms the latter in most of the cases in Table IV.

With scratch training, SeqSleepNet+ results in an average accuracy gain of 1.7%, 6.6%, and 17.3% over DeepSleep-Net+ on Sleep-EDF-SC, Sleep-EDF-ST, and Surrey-cEEGrid, respectively. This is consistent with the findings from the source domain (i.e. the MASS database) in Table III and in [4]. With transfer learning, SeqSleepNet+ also obtains better performance than DeepSleepNet+ with, improving the overall accuracy by 0.8%, 1.5%, and 7.7% on Sleep-EDF-SC, Sleep-EDF-ST, and Surrey-cEEGrid, respectively. These results suggest that DeepSleepNet+ is harder to train and finetune than SeqSleepNet+, especially when the data is small, partly due to its large model footprint [6] and partly due to its reliance on raw signal inputs. However, the results in Table IV show significant gains obtained by both the finetuned models over their scratch counterparts. On the one hand, averaging over all transfer learning scenarios, finetuning SeqSleepNet+ leads to an absolute accuracy gain of 2.5%, 2.0%, and 1.4% on Sleep-EDF-SC, Sleep-EDF-ST, and Surrey-cEEGrid, respectively. Those gains of DeepSleepNet+ are even larger, reaching 3.4%, 7.1%, and 10.9%, respectively, mainly because of the poor performance of the scratch DeepSleepNet+ on Sleep-EDF-ST and Surrey-cEEGrid. Interestingly, transfer learning helps compensate for the lack of training data, evidenced by the observation that the accuracy on Sleep-EDF-SC achieved by the finetuned SeqSleepNet+ is on par with that of MASS (cf. Table III) even though the number of subjects is ten times smaller. On the other hand, despite the heavy data mismatch in the cross-domain scenario, transferring the information of EEG data in the source domain to EOG data in the target domains still yields significant accuracy gains: 1.0% and 7.4% on average with SeqSleepNet+ and DeepSleepNet+, respectively. Interestingly, with the accuracy consistently around 80% obtained from the secondary EOG via DeepSleepNet+'s transfer learning, it is promising to be used as an alternative for EEG in single-channel sleep staging.

Directly applying the pretrained models in the target domains without finetuning results in suboptimal performance in many cases. Averaging over the same-modality transfer learning scenarios, the pretrained SeqSleepNet+ model with direct transfer obtains an accuracy with 10.3%, 5.7%, and 62.2% lower than those obtained by the finetuned models on Sleep-EDF-SC, Sleep-EDF-ST, and Surrey-cEEGrid, respectively. Those gaps in case of DeepSleepNet+ are 16.8%, 9.1%, and 32.6%, respectively. The direct transfer's results are particularly poor under heavy data mismatch conditions, such as the EEG↦EOG scenario and the EEG↦EEG scenario in Surrey-cEEGrid. It is reasonable as substantial differences in characteristics of the source domain and the target domain cause discrepancy in the feature-learning parts of the pretrained models in the target domain. As a consequence, finetuning is essential. Similar findings are also reflected in the class-wise performance (in terms of MF1) in Table V.

The proposed transfer learning approach also outperforms all previous works and set state-of-the-art performance on all three target databases. On Sleep-EDF-SC, with the accuracies of 84.3% (two-channel EEG·EOG) and 85.2% (single-channel EEG) obtained by the transfer learning based SeqSleepNet+, the system yields absolute accuracy gains of 2.0% and 3.2%

TABLE IV: Performance comparison between the proposed transfer-learning systems, and the baseline systems (i.e. the scratch models and the direct-transfer models, in italic font), and previous works. FT and DT are abbreviated for "finetuning" and "direct transfer", respectively. It should be noted that the comparison may not be completely compatible due to differences in experimental setup: *the transfer learning approach was personalized finetuning; †30 minutes of data (mainly Wake epochs) before and after in-bed duration were included, therefore, the results are likely biased towards Wake stage; ‡the evaluation was not subject-independent [5]; ◇the number of subjects is different from that in our experiments.

System	Transfer learning	EEG·EOG·EMG ↦ EEG·EOG·EMG			EEG·EOG↦EEG·EOG			EEG↦EEG			EOG↦EOG			EEG↦EOG		
		Acc.	MF1	κ	Acc.	MF1	κ	Acc.	MF1	κ	Acc.	MF1	κ	Acc.	MF1	κ
Sleep-EDF-SC																
FT SeqSleepNet+	Yes	–	–	–	**84.3**	**77.7**	**0.776**	**85.2**	**79.6**	**0.789**	**81.7**	**75.1**	**0.737**	**80.0**	**72.3**	**0.709**
FT DeepSleepNet+	Yes	–	–	–	**84.6**	**79.0**	**0.782**	**84.4**	**78.8**	**0.781**	**79.8**	**73.4**	**0.713**	**79.4**	**72.8**	**0.707**
DT SeqSleepNet+	Yes	–	–	–	72.0	62.1	0.601	81.2	74.6	0.733	67.2	59.1	0.530	51.1	42.5	0.300
DT DeepSleepNet+	Yes	–	–	–	70.2	59.8	0.586	74.2	66.9	0.651	54.1	41.9	0.396	39.7	35.8	0.235
Scratch SeqSleepNet+	No	–	–	–	82.2	74.2	0.744	82.2	74.1	0.746	78.5	68.3	0.688	78.5	68.3	0.688
Scratch DeepSleepNet+	No	–	–	–	81.9	75.2	0.744	80.8	74.2	0.731	75.9	66.9	0.652	75.9	66.9	0.652
Personalized Deep CNN* [9]	Yes	–	–	–	84.0	–	–	–	–	–	–	–	–	–	–	–
VGG-FT [18]	Yes	–	–	–	–	–	–	80.3	–	–	–	–	–	–	–	–
VGG-FE [18]	Yes	–	–	–	–	–	–	76.3	–	–	–	–	–	–	–	–
ResNet [37]	Yes	76.8	–	–	–	–	–	–	–	–	–	–	–	–	–	–
U-time [61]†	No	–	–	–	–	–	–	–	79.0	–	–	–	–	–	–	–
IITNet [62]†	No	–	–	–	–	–	–	84.0	77.7	0.78	–	–	–	–	–	–
DeepSleepNet† [6]	No	–	–	–	–	–	–	82.0	76.9	0.760	–	–	–	–	–	–
Multitask 1-max CNN [5]	No	–	–	–	82.3	74.7	0.750	81.9	73.8	0.740	–	–	–	–	–	–
1-max CNN [17]	No	–	–	–	–	–	–	79.8	72.0	0.720	–	–	–	–	–	–
Attentional RNN [25]	No	–	–	–	–	–	–	79.1	69.8	0.700	–	–	–	–	–	–
Deep auto-encoder [41]	No	–	–	–	–	–	–	78.9	73.3	–	–	–	–	–	–	–
Deep CNN [7]	No	–	–	–	–	–	–	74.8	69.8	–	–	–	–	–	–	–
Decision trees [63]‡	No	–	–	–	–	–	–	93.1	–	–	–	–	–	–	–	–
k-NN [64]‡	No	–	–	–	–	–	–	80.0	–	–	–	–	–	–	–	–
GMM [65]‡	No	–	–	–	–	–	–	73.3	–	–	–	–	–	–	–	–
Sleep-EDF-ST																
FT SeqSleepNet+	Yes	**80.6**	**76.2**	**0.727**	**81.0**	**76.7**	**0.732**	**81.0**	**77.5**	**0.734**	**80.4**	**76.5**	**0.722**	**79.6**	**75.2**	**0.710**
FT DeepSleepNet+	Yes	**80.2**	**76.6**	**0.722**	**80.1**	**76.6**	**0.721**	**81.5**	**77.5**	**0.738**	**77.4**	**74.1**	**0.682**	**76.0**	**71.4**	**0.661**
DT SeqSleepNet+	Yes	79.3	73.0	0.703	73.1	64.2	0.615	80.5	75.6	0.722	67.2	59.4	0.531	56.3	48.4	0.363
DT DeepSleepNet+	Yes	74.6	67.4	0.645	71.6	65.4	0.600	66.7	61.3	0.541	70.0	63.3	0.586	35.1	31.0	0.116
Scratch SeqSleepNet+	No	79.4	74.5	0.709	79.6	74.8	0.711	76.5	70.6	0.667	78.6	71.6	0.693	78.6	71.6	0.693
Scratch DeepSleepNet+	No	73.8	69.6	0.634	73.7	67.6	0.629	72.4	64.6	0.603	70.0	65.9	0.574	70.0	65.9	0.574
SVM + Scattering Trans. [66]	No	–	–	–	–	–	–	78.6	73.6	0.695	–	–	–	–	–	–
Decision trees. [67]	No	–	–	–	–	–	–	75.0	–	–	–	–	–	–	–	–
Surrey-cEEGrid																
FT SeqSleepNet+	Yes	**82.9**	**72.6**	**0.762**	**82.3**	**71.1**	**0.752**	**75.3**	**60.8**	**0.650**	**82.6**	**72.2**	**0.758**	**81.9**	**71.2**	**0.749**
FT DeepSleepNet+	Yes	71.1	59.7	0.588	**77.8**	**66.5**	**0.687**	58.2	42.8	0.391	77.5	66.6	0.682	81.7	70.5	0.745
DT SeqSleepNet+	Yes	20.2	14.7	0.062	19.4	14.6	0.051	10.6	9.1	−0.015	24.3		0.085	24.1	16.9	0.090
DT DeepSleepNet+	Yes	38.4	11.8	0.025	38.3	11.7	0.020	38.4	11.6	0.012	39.3	25.4	0.214	38.9	25.4	0.195
Scratch SeqSleepNet+	No	82.1	67.6	0.748	81.5	66.4	0.739	71.9	55.2	0.597	81.3	67.8	0.737	81.3	67.8	0.737
Scratch DeepSleepNet+	No	65.6	57.3	0.535	65.4	57.4	0.534	42.5	30.3	0.195	69.1	60.0	0.579	69.1	60.0	0.579
Random Forests◇ [11]	No	–	–	–	72.0	–	0.600	70.0	–	0.580	–	–	–	–	–	–

TABLE V: Class-wise performance of the proposed transfer-learning systems and the baseline systems in terms of MF1.

System	Transfer learning	EEG·EOG·EMG↦EEG·EOG·EMG					EEG·EOG↦EEG·EOG					EEG↦EEG					EOG↦EOG					EEG↦EOG				
		W	N1	N2	N3	REM	W	N1	N2	N3	REM	W	N1	N2	N3	REM	W	N1	N2	N3	REM	W	N1	N2	N3	REM
Sleep-EDF-SC																										
FT SeqSleepNet+	Yes	–	–	–	–	–	80.0	45.9	88.0	85.9	88.9	85.4	50.9	88.8	86.4	86.5	75.1	46.4	86.3	80.3	87.3	72.8	40.3	84.9	78.7	84.8
FT DeepSleepNet+	Yes	–	–	–	–	–	82.6	50.0	87.8	86.2	88.4	81.0	50.5	88.2	86.9	87.2	75.3	42.7	84.5	79.3	85.4	75.7	41.9	83.9	78.1	84.5
DT SeqSleepNet+	Yes	–	–	–	–	–	63.2	29.8	84.9	72.2	60.2	74.1	46.9	86.9	81.2	83.8	67.7	33.9	79.3	54.4	60.4	51.4	28.9	61.0	19.8	51.6
DT DeepSleepNet+	Yes	–	–	–	–	–	59.6	30.6	82.7	80.5	45.5	69.0	31.9	80.0	74.6	78.8	38.9	15.1	74.6	78.8	2.0	29.6	11.5	48.9	75.1	13.8
Scratch SeqSleepNet+	No	–	–	–	–	–	75.0	38.3	86.8	86.0	85.0	78.5	37.1	87.6	86.2	81.2	73.5	25.8	84.4	77.7	80.3	73.5	25.8	84.4	77.7	80.3
Scratch DeepSleepNet+	No	–	–	–	–	–	67.5	47.9	86.8	86.8	87.0	70.3	48.1	86.4	84.6	81.3	62.8	33.1	81.5	74.8	82.5	62.8	33.1	81.5	74.8	82.5
Sleep-EDF-ST																										
FT SeqSleepNet+	Yes	80.5	54.0	84.2	71.9	91.1	81.0	55.5	84.7	71.8	90.4	81.8	59.5	84.4	72.9	89.2	80.3	57.7	83.9	70.4	90.3	78.9	52.7	83.5	71.8	88.9
FT DeepSleepNet+	Yes	80.8	55.7	82.9	71.3	87.9	81.7	57.3	83.8	70.7	89.5	82.9	56.9	85.2	74.0	88.4	81.8	56.1	81.3	63.8	87.3	77.0	45.7	80.7	68.5	85.3
DT SeqSleepNet+	Yes	76.9	47.5	85.4	68.5	86.7	63.5	39.6	85.2	65.5	67.3	78.8	54.0	85.2	69.2	88.6	62.2	44.0	80.4	61.9	48.3	62.8	41.1	66.5	17.7	53.8
DT DeepSleepNet+	Yes	67.9	33.7	81.9	71.9	81.6	73.2	36.0	78.4	70.2	69.3	66.8	36.1	73.6	63.4	66.6	62.4	32.8	79.6	70.5	71.2	32.4	13.7	40.6	64.1	4.3
Scratch SeqSleepNet+	No	80.9	49.2	83.5	70.0	89.1	82.0	48.2	83.3	70.7	89.6	80.3	38.9	82.0	69.9	81.6	76.5	38.4	83.6	72.2	87.2	76.5	38.4	83.6	72.2	87.2
Scratch DeepSleepNet+	No	72.0	46.1	78.3	67.4	84.4	59.5	47.4	80.9	72.3	77.8	61.0	40.3	81.1	67.1	73.7	67.1	45.7	74.9	64.3	77.2	67.1	45.7	74.9	64.3	77.2
Surrey-cEEGrid																										
FT SeqSleepNet+	Yes	91.6	81.3	27.2	81.3	81.3	90.9	79.9	23.1	81.4	80.2	76.0	58.0	10.6	71.5	73.4	91.2	78.8	26.6	81.1	83.3	91.3	76.0	26.4	81.0	81.0
FT DeepSleepNet+	Yes	80.4	63.3	10.3	67.5	77.1	86.7	70.9	15.5	77.1	82.5	74.9	23.2	5.7	47.0	63.4	87.0	66.6	22.0	77.7	79.8	90.2	78.5	20.2	80.9	82.8
DT SeqSleepNet+	Yes	57.0	1.6	11.9	3.1	0.0	54.4	0.4	12.1	6.0	0.0	29.2	0.6	9.9	3.3	2.5	46.7	12.1	13.7	30.1	0.0	67.9	2.8	10.1	3.7	0.1
DT DeepSleepNet+	Yes	57.6	0.0	1.3	0.0	0.0	57.0	0.0	1.7	0.0	0.0	57.1	1.2	0.0	0.0	0.0	67.0	0.4	17.2	42.4	0.0	66.2	3.1	14.6	43.3	0.0
Scratch SeqSleepNet+	No	90.8	81.1	5.7	80.2	80.0	90.4	79.9	3.3	79.9	78.7	88.2	50.5	1.0	67.4	68.8	90.3	79.6	12.3	79.5	77.5	90.3	79.6	12.3	79.5	77.5
Scratch DeepSleepNet+	No	75.0	61.4	23.4	64.4	62.4	72.9	59.4	24.0	68.7	61.8	57.6	12.3	6.9	23.1	51.4	75.2	66.1	19.7	73.0	66.0	75.2	66.1	19.7	73.0	66.0

over the best non-transfer-learning systems, Multitask 1-max CNN [5] (82.3%) and DeepSleepNet [6] (82.0%), respectively. Those respective gains achieved by the transfer learning based DeepSleepNet+ are 2.3% and 2.4%. Large margins, 7.5% and 7.8%, are seen when contrasting the proposed SeqSleepNet+ and DeepSleepNet+ systems with the existing transfer learning approach based on ResNet [37] and VGGNet [18]. These results suggest that the quality of the base model plays an important role in transfer learning for sleep staging. The results obtained by the proposed systems are also better than the personalization results in [9] even though cohort transfer learning here is more challenging than personalized transfer learning as, with the former, we do not have access to test subjects' data during training. Similar to Sleep-EDF-SC, both proposed systems are superior to previous works on Sleep-EDF-ST. However, on Surrey-cEEGrid, while the transfer learning based SeqSleepNet+ uplifts the accuracy by a margin of 10.3% in two-channel EEG·EOG and 5.3% in

single-channel EEG compared to the seminal work in [11], the DeepSleepNet+ experiences an accuracy drop of 11.8% in single-channel EEG even though 5.8% absolute accuracy gain is seen in two-channel EEG·EOG.

4) Influence of the number of finetuning subjects: This section investigates the influence of the amount of the target-domain data to the network finetuning. Considering the EEG·EOG↦EEG·EOG scenario and the entire-network finetuning strategies for this investigation. For a target domain, we randomly selected 25% of the subjects as the test subjects while the remaining subjects were used for finetuning. A pretrained network was finetuned using data from the finetuning set of N subjects for 500 finetuning steps and the test accuracy was recorded during the finetuning process. Starting with the finetuning set of $N = 1$ subject, we repeated this procedure and added two more subjects into it at each iteration.

Figure 8 shows the learning curves recorded with varying number of finetuning subjects. The learning curves present a strong impact of the number of finetuning subjects on SeqSleepNet+ while such influence on DeepSleepNet+ is less noticeable, except for Surrey-cEEGrid. It is rational if these results are linked to the networks' finetuning behaviors. While a pretrained SeqSleepNet+ requires its feature-learning parts to be adapted into the target domains, this requirement is not mandatory for DeepSleepNet+, except for the cEEGrid data (see Section V-C.2). And when the feature-learning parts need to be adjusted, less finetuning data make the networks converge to more subject-specific solutions, i.e. overfitting. On the contrary, more finetuning data allow the feature learning parts to converge to more generalizable solutions. This is supported by the SeqSleepNet+'s learning curves on the Sleep-EDF-SC and Surrey-cEEGrid domain, and DeepSleepNet+'s learning curves on the Surrey-cEEGrid domain. From these curves, we also speculate that when the feature-learning parts of a network needs to be adapted to a target domain, a generalizable solution can be obtained with the number of finetuning subjects being around 11-13. Particularly, the learning curves on Sleep-EDF-ST appears to be counter-intuitive as more finetuning subjects occasionally result in lowering learning curves. These irregularities can be explained by the fact that the Sleep-EDF-ST population has a very wide range of age, 18-79. As sleep patterns change with age [68], depending the age range of the test subjects, including a subject whose age is far from that range would hurt more than help. Further studies how to determine and select candidates from a population that are most beneficial for a finetuning task.

D. Discussion

It is worth mentioning that, although we focused on studying with small cohorts in this work, the presented transfer learning approach would also be useful for a sleep study with a larger cohort. On the one hand, it only requires the data of a handful of subjects to be labelled, avoiding the burden of manual scoring the entire cohort. On the other hand, finetuning a pretrained model is generally much faster than training a model from scratch, as illustrated in Figure 7. This is because the pretrained model has reached already a reasonable accuracy. As a result, it is able to converge after a few
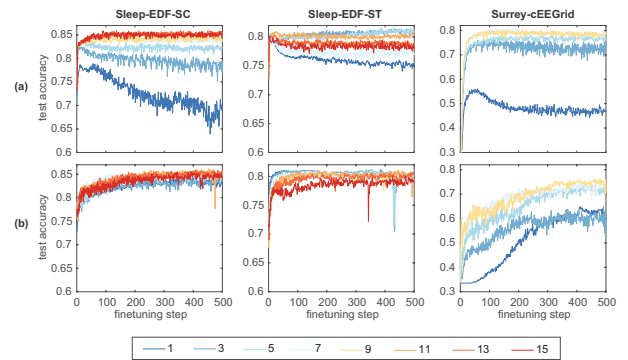

Fig. 8: The learning curves obtained from the test data with varying the number of finetuning subjects. (a) SeqSleepNet+ and (b) DeepSleepNet+.

additional finetuning epochs. On the downside, it is worth noting that still data from a number of subjects is needed for the validation purpose and future works should explore regularization methods, such as Kullback–Leibler divergence [69], to eliminate this requirement.

VI. CONCLUSION

We presented a deep transfer learning approach to address the problem of insufficient data in many sleep studies and to improve automatic sleep staging performance on small cohorts. The SeqSleepNet+ and DeepSleepNet+ derived from the presented generic sequence-to-sequence sleep staging framework were employed to surpass data mismatch and enable transferring information from the source domain to the target domain. The networks were trained in the source domain and then finetuned in the target domains to complete knowledge transfer. Experiments were conducted with different finetuning strategies, transfer learning scenarios, and target domains. The experimental results showed that via transfer learning, the sleep staging performance was significantly improved across all learning cases over the scratch models trained solely on the target domains. The results also revealed the different behaviors of two SeqSleepNet+ and DeepSleepNet+ models in transfer learning. The former was found more consistent and stable and outperformed the latter in most of the transfer learning experiments. The number of subjects required for finetuning also varied between the two networks, however, overall, a small number of finetuning subjects was needed for the networks to converge to a generalizable solution.

REFERENCES

[1] 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.

[2] 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, pp. 644, 1969.

[3] S. J. Redmond and C. Heneghan, "Cardiorespiratory-based sleep staging in subjects with obstructive sleep apnea," *IEEE Trans Biomed Eng*, vol. 53, pp. 485–496, 2006.

[4] H. Phan *et al.*, "SeqSleepNet: end-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging," *IEEE Trans Neural Syst Rehabil Eng*, vol. 27, no. 3, pp. 400–410, 2019.

[5] H. Phan *et al.*, "Joint classification and prediction CNN framework for automatic sleep stage classification," *IEEE Trans Biomed Eng*, vol. 66, no. 5, pp. 1285–1296, 2019.

[6] A. Supratak *et al.*, "DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG," *IEEE Trans Neural Syst Rehabil Eng*, vol. 25, no. 11, pp. 1998–2008, 2017.

[7] O. Tsinalis *et al.*, "Automatic sleep stage scoring with single-channel EEG using convolutional neural networks," *arXiv:1610.01683*, 2016.

[8] J. B. Stephansen *et al.*, "Neural network analysis of sleep stages enables efficient diagnosis of narcolepsy," *Nat. Commun.*, vol. 9, no. 1, 2018.

[9] K. Mikkelsen and M. De Vos, "Personalizing deep learning models for automatic sleep staging," *arXiv Preprint arXiv:1801.02645*, 2018.

[10] S. Chambon *et al.*, "A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series," *IEEE Trans Neural Syst Rehabil Eng*, vol. 26, no. 4, pp. 758–769, 2018.

[11] K. B. Mikkelsen *et al.*, "Machine-learning-derived sleep–wake staging from around-the-ear electroencephalogram outperforms manual scoring and actigraphy," *J Sleep Res*, vol. 28, no. 2, pp. e12786, 2019.

[12] D. Looney *et al.*, "Wearable in-ear encephalography sensor for monitoring sleep. preliminary observations from nap studies," *Ann Am Thorac Soc*, vol. 13, no. 12, pp. 32–42, 2016.

[13] V. Goverdovsky *et al.*, "In-ear EEG from viscoelastic generic earpieces: Robust and unobtrusive 24/7 monitoring," *IEEE Sens J*, vol. 16, pp. 271–277, 2016.

[14] P. Kidmose *et al.*, "A study of evoked potentials from ear-EEG," *IEEE Trans Biomed Eng*, vol. 60, no. 10, pp. 2824–2830, 2013.

[15] H. Dong *et al.*, "Mixed neural network approach for temporal sleep stage classification," *IEEE Trans Neural Syst Rehabil Eng*, vol. 26, no. 2, pp. 324–333, 2018.

[16] F. Andreotti, H. Phan, and M. De Vos, "Visualising convolutional neural network decisions in automatic sleep scoring," in *Proc. AIH*, 2018.

[17] H. Phan *et al.*, "DNN filter bank improves 1-max pooling CNN for single-channel EEG automatic sleep stage classification," in *Proc. EMBC*, 2018, pp. 453–456.

[18] A. Vilamala *et al.*, "Deep convolutional neural networks for interpretable analysis of EEG sleep stage scoring," in *Proc. MLSP*, 2017.

[19] A. Sors *et al.*, "A convolutional neural network for sleep stage scoring from raw single-channel eeg," *Biomed Signal Process Control*, vol. 42, pp. 107–114, 2018.

[20] A. Patanaik, J. L. Ong, J. J. Gooley, S. Ancoli-Israel, and M. W. L. Chee, "An end-to-end framework for real-time automatic sleep stage classification," *Sleep*, vol. 41, no. 5, pp. zsy041, 2018.

[21] S. Biswal *et al.*, "Expert-level sleep scoring with deep neural networks," *J Am Med Inform Assoc.*, 2018.

[22] S. Biswal *et al.*, "SLEEPNET: Automated sleep staging system via deep learning," *arXiv preprint arXiv:1707.08262*, 2017.

[23] H. Sun *et al.*, "Large-scale automated sleep staging," *SLEEP*, vol. 40, no. 10, pp. zsx139, 2017.

[24] A. N. Olesen *et al.*, "Deep residual networks for automatic sleep stage classification of raw polysomnographic waveforms," in *Proc. EMBC*, 2018, pp. 3713–3716.

[25] H. Phan *et al.*, "Automatic sleep stage classification using single-channel eeg: Learning sequential features with attention-based recurrent neural networks," in *Proc. EMBC*, 2018, pp. 1452–1455.

[26] P. Koch *et al.*, "Recurrent neural network based early prediction of future hand movements," in *Proc. EMBC*, 2018, pp. 4710–4713.

[27] P. Koch *et al.*, "Recurrent neural networks with weighting loss for early prediction of hand movements," in *Proc. EUSIPCO*, 2018.

[28] B. Zhai *et al.*, "Making sense of sleep: Multimodal sleep stage classification in a large, diverse population using movement and cardiac sensing," in *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2020, vol. 4.

[29] O. Faust *et al.*, "A review of automated sleep stage scoring based on physiological signals for the new millennia," *Comput Methods Programs Biomed*, vol. 176, pp. 81–91, 2019.

[30] Y. Roy *et al.*, "Deep learning-based electroencephalography analysis: a systematic review," *J Neural Eng*, vol. 16, no. 5, 2019.

[31] L. Fiorillo *et al.*, "Automated sleep scoring: A review of the latest approaches," *Sleep Medicine Reviews*, vol. 48, pp. 101204, 2019.

[32] N. Cooray *et al.*, "Detection of REM sleep behaviour disorder by automated polysomnography analysis," *Clin Neurophysiol.*, vol. 130, no. 4, pp. 505–514, 2019.

[33] H. Phan *et al.*, "Deep transfer learning for single-channel automatic sleep staging with channel mismatch," in *Proc. EUSIPCO*, 2019.

[34] B. Kemp *et al.*, "Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG," *IEEE Trans Biomed Eng*, vol. 47, no. 9, pp. 1185–1194, 2000.

[35] 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.

[36] A. N. Olesen *et al.*, "A noise-assisted data analysis method for automatic eog-based sleepstage classification using ensemble learning," in *Proc. EMBC*, 2016, pp. 3769–3772.

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

[38] H. Phan *et al.*, "Fusion of end-to-end deep learning models for sequence-to-sequence sleep staging," in *Proc. EMBC*, 2019.

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

[40] A. Sterr *et al.*, "Sleep eeg derived from behind-the-ear electrodes (ceegrid) compared to standard polysomnography: A proof of concept study," *Front Hum Neurosci*, vol. 12, no. 452, 2018.

[41] O. Tsinalis, P. M. Matthews, and Y. Guo, "Automatic sleep stage scoring using time-frequency analysis and stacked sparse autoencoders," *Ann Biomed Eng*, vol. 44, no. 5, pp. 1587–1597, 2016.

[42] S. A. Imtiaz and E. Rodriguez-Villegas, "Recommendations for performance assessment of automatic sleep staging algorithms," in *Proc. EMBC*, 2014, pp. 5044–5047.

[43] S. A. Imtiaz and E. Rodriguez-Villegas, "An open-source toolbox for standardized use of PhysioNet Sleep EDF Expanded Database," in *Proc. EMBC*, 2015, pp. 6014–6017.

[44] S. Debener *et al.*, "Unobtrusive ambulatory EEG using a smartphone and flexible printed electrodes around the ear," *Sci. Rep.*, vol. 5, pp. 16743, 2015.

[45] S. Debener *et al.*, "How about taking a low-cost, small, and wireless eeg for a walk?," *Psychophysiology*, vol. 49, no. 11, 2012.

[46] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT press, 2016.

[47] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[48] M. Schuster and K. P. Kuldip, "Bidirectional recurrent neural networks," *IEEE Trans. on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[49] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computing*, vol. 9, no. 8, pp. 1735–1780, 1997.

[50] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. EMNLP*, 2014.

[51] T. Cooijmans *et al.*, "Recurrent batch normalization," *arXiv Preprint arXiv:1603.09025*, 2016.

[52] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015, pp. 448–456.

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

[54] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans Knowl Data Eng*, vol. 22, no. 10, pp. 1345–1359, 2010.

[55] B. Koley and D. Dey, "An ensemble system for automatic sleep stage classification using single channel EEG signal," *Comput Biol Med*, vol. 42, no. 12, pp. 1186–95, 2012.

[56] 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.

[57] H. Phan *et al.*, "Metric learning for automatic sleep stage classification," in *Proc. EMBC*, 2013, pp. 5025–5028.

[58] S.-F. Liang *et al.*, "Development of an EOG-based automatic sleep-monitoring eye mask," *IEEE Trans Instrum Meas*, vol. 64, no. 11, pp. 2977–2985, 2015.

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

[60] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–13.

[61] M. Perslev *et al.*, "U-time: A fully convolutional network for time series segmentation applied to sleep staging," in *Proc. NeurIPS*, 2019.

[62] S. Back *et al.*, "Intra- and inter-epoch temporal context network (IITNet) for automatic sleep stage scoring," *arXiv:1902.06562*, 2019.

[63] K. A. I. Aboalayon *et al.*, "Sleep stage classification using EEG signal analysis: A comprehensive survey and new investigation," *Entropy*, vol. 18, no. 9, pp. 272, 2016.

[64] J. L. Rodríguez-Sotelo *et al.*, "Automatic sleep stages classification using eeg entropy features and unsupervised pattern analysis techniques," *Entropy*, vol. 16, no. 12, pp. 6573–6589, 2014.

[65] A. M. Munk *et al.*, "Semi-supervised sleep-stage scoring based on single channel EEG," in *Proc. ICASSP*, 2018.

[66] G.-R. Liu *et al.*, "Diffuse to fuse EEG spectra – intrinsic geometry of sleep dynamics for classification," *arXiv:1803.01710*, 2019.

[67] T. H. Sanders, M. McCurry, and M. A. Clements, "Sleep stage classification with cross frequency coupling," in *Proc. EMBC*, 2014.

[68] A. C. Skeldon, G. Derks, and D.-J. Dijk, "Modelling changes in sleep timing and duration across the lifespan: Changes in circadian rhythmicity or sleep homeostasis?," *Sleep Med Rev*, vol. 28, pp. 96–107, 2016.

[69] H. Phan *et al.*, "Personalized automatic sleep staging with single-night data: a pilot study with KL-divergence regularization," *Physiological Measurement*, vol. 41, no. 6, pp. 064004, 2020.