Transferring Structured Knowledge in Unsupervised Domain Adaptation of a Sleep Staging Network

Chaehwa Yoo, Hyang Woon Lee, and Je-Won Kang, *Member, IEEE*

***Abstract*—Automatic sleep staging based on deep learning (DL) has been attracting attention for analyzing sleep quality and determining treatment effects. It is challenging to acquire long-term sleep data from numerous subjects and manually labeling them even though most DL-based models are trained using large-scale sleep data to provide state-of-the-art performance. One way to overcome this data shortage is to create a pre-trained network with an existing large-scale dataset (source domain) that is applicable to small cohorts of datasets (target domain); however, discrepancies in data distribution between the domains prevent successful refinement of this approach. In this paper, we propose an unsupervised domain adaptation method for sleep staging networks to reduce discrepancies by re-aligning the domains in the same space and producing domain-invariant features. Specifically, in addition to a classical domain discriminator, we introduce local discriminators - *subject and stage* - to maintain the intrinsic structure of sleep data to decrease local misalignments while using adversarial learning to play a minimax game between the feature extractor and discriminators. Moreover, we present several optimization schemes during training because the conventional adversarial learning is not effective to our training scheme. We evaluate the performance of the proposed method by examining the staging performances of a baseline network compared with direct transfer (DT) learning in various conditions. The experimental results demonstrate that the proposed domain adaptation significantly improves the performance though it needs no labeled sleep data in target domain.**

***Index Terms*—Sleep staging, unsupervised domain adaptation, local alignment, knowledge transfer.**

Manuscript received February 5, 2021; revised May 27, 2021 and July 14, 2021; accepted August 4, 2021. Date of publication August 13, 2021; date of current version March 7, 2022. This work was supported by Convergent Technology R&D Program for Human Augmentation through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT under Grants NRF-2019M3C1B8090803 and NRF-2019M3C1B8090804 and in part by the Ewha Womans University scholarship of 2020. *(Corresponding author: Je-Won Kang.)*

Chaehwa Yoo and Je-Won Kang are with the Department of Electronic and Electrical Engineering and Graduate Program in Smart Factory, Ewha Womans University, Seoul 03760, Korea (e-mail: chyoo@ewhain.net; jewonk@ewha.ac.kr).

Hyang Woon Lee is with the Departments of Neurology, Medical Science, Computational Medicine, College of Medicine, and Graduate Program in System Health Science and Engineering, Ewha Womans University, Seoul 07985, Korea (e-mail: leeh@ewha.ac.kr).

Digital Object Identifier 10.1109/JBHI.2021.3103614

I. Introduction

SLEEP staging refers to the identification of the current stage of sleep using biometric signals [1]. Because sleep staging can help explain sleep disorders to provide effective treatments, researchers have developed accurate sleep scoring methods [2], [3]. Sleep has a basic structural organization, with five unique stages, including "wake," "rapid eye movement (REM)," and "non-rapid eye movement (NREM)"; NREM is generally separated into three stages as NREM 1 (N1), NREM 2 (N2), and NREM 3 (N3) [2]. Sleep experts manually evaluate sleep stages in compliance with standard guidelines, traditionally based on polysomnography (PSG) data, to obtain the ground truth [4], [5]. PSG is a comprehensive test to diagnose sleep disorders or to evaluate treatment effects by monitoring various biometric signals, such as electroencephalogram (EEG), electrooculogram (EOG), and electromyogram (EMG) [6], which can be used to categorize sleep samples into one of the five stages as mentioned above.

Unfortunately, manual evaluations are expensive and vulnerable to human errors [3], [7]–[9]. Sleep experts examine 30 s segments of data called epochs and repeat the examinations for the entire sleep cycle [3], [7]–[9]. Even experienced experts can evaluate data with only 90 percent accuracy owing to heavy workloads. Therefore, automated sleep staging has gained substantial interest [2], [3]. Early studies used handcrafted features to reflect the physical significance of biometric signals; these works have also attempted to use classical machine learning methods to classify sleep stages. Recent advances in artificial intelligence have led to deep learning (DL)-based sleep staging studies [9]–[22]. Convolutional neural networks (CNNs) are used to extract sleep features, and recurrent neural networks (RNNs) are used to extract features by analyzing the relationships among adjacent epochs [10], [11], [17], [20]–[22].

Developments in recent DL models for sleep staging have been facilitated by large-scaled open-access sleep datasets, such as MASS dataset [23]. The most straightforward model training approach is to pre-train a model with a large-scale dataset and enable its application via finetuning with small-scale datasets, developed for investigating specific sleep disorders or exploring the feasibilities of new monitoring devices [12], [24], [25]. However, this refinement cannot provide reliable performance because the training and testing samples are not drawn from the same probability distribution. Even the large-scale dataset is

not guaranteed to contain enough sleep samples from various subjects because the biomedical data are almost impossible to disclose. Further, the recording conditions are significantly different from one device to another. While EEG signals are recorded during PSG tests using accurate equipment, consumer-level wearable devices may be prone to undesirable noise under local test conditions.

In this paper, we propose an efficient training method for domain adaptation (DA) for sleep staging, thereby helping a model maintain its performance when testing with samples from a different dataset. The approach attempts to alleviate the undesired effects of domain discrepancies. However, in practical scenarios, whereas both sleep data and ground truth labels are available for training in source domain, it is difficult to obtain labeled sleep data in target domain to which a model would be deployed. To challenge this problem, we consider an unsupervised DA setting, inspired by [26], where both the source and target data are available for training but only the source is labeled. Specifically, a large-scale sleep dataset in source aids training of small-scale unlabeled sleep data in target by transferring knowledge from the source. This consideration is more consistent with realistic and practical circumstances, which significantly differs from the supervised DA settings in previous works [27]–[29].

Our DL model is trained using adversarial learning, in which a feature extractor and a discriminator compete with each other, to deceive a domain boundary and overcome the discrepancies between the domains, as in [26]. We attempt to transfer the knowledge from a source containing an organized structure of sleep data instead of adopting a straightforward adversarial training method. Specifically, we use not only a domain discriminator to identify the domain of the input samples but also other key discriminators to reflect the intrinsic relationships between the source and target sleep data. This training strategy helps maintain the internal and local structures of both domain distributions and provides more reliable results from among various datasets and acquisition environments. Our model is optimized using several training schemes to improve sleep staging performance and produce domain-invariant features simultaneously. The proposed approach can address the time and labor involved in sleep data acquisition and have great impact on sleep monitoring and analyses. We highlight main contributions as follows:

- We propose an efficient unsupervised DA training strategy for sleep staging networks to overcome domain discrepancies between source domain and target domain in more realistic circumstances and produce domain-invariant features, by introducing local discriminators to maintain the intrinsic structure of sleep data.
- We present several optimization schemes used for stage-wise and subject-wise discriminators as local discriminators to achieve both the global and local DA during training and improve classification performance.
- We conduct comprehensive experiments to display improved performance by the proposed method in various DL models, datasets, PSG montages, and configurations, while there are few DA studies for sleep data. The proposed method provides significantly improved performance although it uses unlabeled sleep samples in target domain.

II. Related Works

A. Sleep Staging Analysis

DL has been actively applied to the analyses of sleep data using different architectures, such as autoencoder [13], deep neural networks (DNNs) [14], CNNs [9], [15], and RNNs [16]. Different types of input signals such as spectrogram [11], EOG [12], ECG [17], and photoplethysmogram (PPG) [18] signals are used in addition to an EEG signal. Current DL-based sleep staging networks are generally composed of two parts: feature extraction and sequence signal classification. DeepSleepNet [10] was proposed as a network architecture to use CNNs to extract both temporal and spatial features from an epoch, and long short-term memory (LSTM) was subsequently applied to learn the temporal relations between adjacent epochs.

Various DL methods have been proposed to combine handcrafted features as complements. Sun *et al.* [19] and Dong *et al.* [14] proposed to use hierarchical features of handcrafted features and learned features from time series of epochs for optimizing classification performance. Qu *et al.* [20] proposed a multi-scale deep architecture by decomposing an EEG signal into different frequency bands at the input; they also utilized the multi-head self-attention module of the transformer model to model the global temporal context.

There are several works on enhancing performance by exploiting sequential learning instead of one or few epochs. Phan *et al.* [21] used contextual outputs of neighboring epochs to help decide the current sleep staging. SeqSleepNet [11] considered a sequence of several epochs for an input and classified all the labels simultaneously; two different RNNs were employed to learn the short-term and long-term features of sequential epochs in different time windows. Seo *et al.* [22] introduced the intra- and inter-epoch temporal context network using sub-epoch features.

B. Domain Adaptation

DA was developed to mitigate problems when the probability distribution of the source was substantially different from that of the target. Depending on the degree of labeled data in the target domain, the approach is categorized as supervised [30], semi-supervised [31], or unsupervised DA [32]. One popular approach is to train the deep features using an adversarial loss to confuse a domain classifier and produce domain-invariant features [26], [33]. Several studies have considered conditional probability distributions in addition to marginal distributions. Specifically, a multi-adversarial domain adaptation (MADA) [34] used an adversarial learning module for an individual class to improve performance with the target. In [32], a pseudo-label was endorsed for an unlabeled sample in the target and jointly trained with a real label in the source. These methods have attempted to align the discrepancies both in the domain-level distributions as global adaptation and category-level distributions as local adaptation. Such alignments were individually carried out.

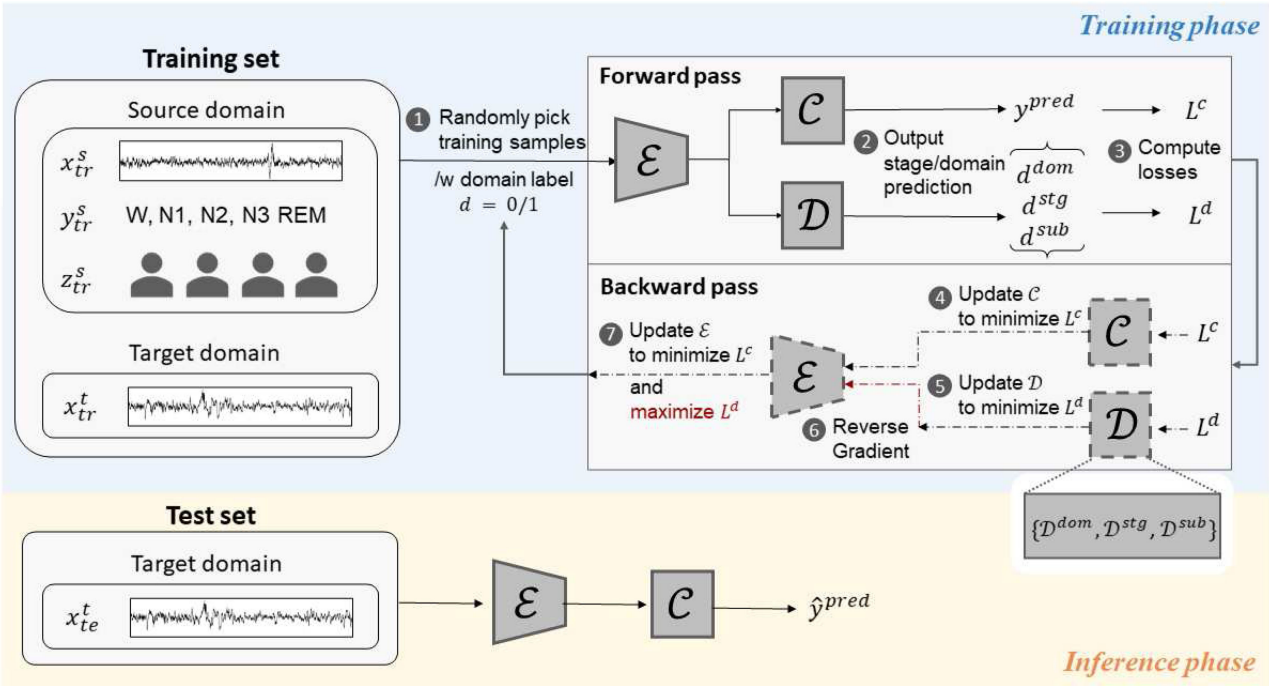

Fig. 1. The training framework using the proposed domain adaptation. In the training, a feature extractor is trained to produce domain-invariant features by transferring knowledge from source to target. The network achieves the goal via a minimax game between the feature extractor and discriminators without labeled samples in the target domain. In the inference, the sleep stage of an input sample in a target domain is predicted using the knowledge transferred from the source domain.

However, the training scheme was suboptimal since the global and local domain-shifts were not often made in the same directions. In [35], the GSDA was applied to align the local and global distributions consistently by synchronizing training.

There has been increasing interest in developing learning schemes for biomedical data to overcome the limited quality and quantity of the training samples [36], [37]. Gu *et al.* [36] adopted adversarial learning to perform an attribute-invariant translation and improve the performance for skin disease recognition. Zhang *et al.* [37] proposed an unsupervised cross-subject adaptation method to predict the locomotion intent of an unlabeled target subject. For sleep staging, several works have attempted to train personalized DL models [8]. Subjects with sleep disorders [38] or within certain age ranges [13], [15] have been considered for personalization. However, these approaches need to consider more labeled data and acquisition conditions in the target. To overcome these problems, Phan *et al.* [27] proposed deep transfer learning to address data-variability and data-inefficiency issues. Jaoude *et al.* [28] also used a similar transfer learning approach with long-term scalp EEG recordings. Banluesombatkul *et al.* [29] proposed a transfer learning framework based on model agnostic meta-learning (MAML). Nasiri *et al.* [39] proposed an adversarial training based method along with attention mechanisms to pay attention to more relevant input channels.

The approach presented here is developed under the consideration of a totally unsupervised setting while resolving the same issues to alleviate performance drop in the target compared with previous studies. Our method does not need a finetuning phase as it uses both source and target samples simultaneously during training. Further, several effective optimizations fitted to local attributes of sleep data are developed to improve the performance.

III. Methodologies

A. Proposed Domain Adaptation in Sleep Staging

We present a training method for a domain-invariant sleep staging network to provide more reliable features both in the source and target. The proposed method aims to enhance the performance in the target despite lacking labels. X^S and X^T denote sleep datasets in the source and target containing n and m samples, respectively. It is noted that a source sample $x^s_i \in \mathbf{X}^S = \{(x^s_i, y^s_i, z^s_i)\}^n_{i=1}$ has a label $y^s_i \in \mathbf{C}^S = \{\text{W, N1, N2, N3, REM}\}$. $z^s_i \in \mathbf{Z}^S = \{S_1, S_2, \ldots, S_b\}$ is an indicator of a subject or a group of subjects in $\mathbf{X}^S$ to specify the subjects x^s_i during training. Such labels have not been used in previous works [27]–[29], [39]. There are m_{tr} training samples and m_{te} testing samples in $\mathbf{X}^T$. The target sample $x^t_j \in \mathbf{X}^T$ is not annotated in our approach.

A network trained using only the source samples is not generally applicable to a target owing to the lack of labels even though n is greater than m. Therefore, the knowledge from the source such as y^s and z^s is transferred to improve the performance in the target using adversarial learning to deceive the domain boundary between the domains. Fig. 1 illustrates the overall training framework of the proposed domain adaptation; it includes a feature extractor $\mathcal{E}$, a set of discriminators $\mathcal{D}$, and

a stage classifier $\mathcal{C}$. In our method, $\mathcal{D} = \{\mathcal{D}^{dom}, \mathcal{D}^{stg}, \mathcal{D}^{sub}\}$ is developed to facilitate the knowledge transfer using the intrinsic structures of sleep data instead of employing the conventional adversarial learning. $\mathcal{D}^{dom}$ is used for global adaptation to create a co-feature space between two domains. $\mathcal{D}^{stg}$ and $\mathcal{D}^{sub}$ are a stage-discriminator and a subject-discriminator. We further use $\mathcal{D}^{stg}$ and $\mathcal{D}^{sub}$ for local adaptation to reflect the internal structures of the sleep data whereas only the $\mathcal{D}^{dom}$ was used in previous works [26], [33]. We explain more details on this in Section III-B.

1) Overall Transfer Learning Scheme: Our network is trained to transfer source knowledge and produce domain-invariant features via a minmax game between $\mathcal{E}$ and $\mathcal{D}$ given a random input sample. The training phase is composed of a forward and a backward pass to train the network as shown in Fig. 1. In the forward pass, a training sample is inputted to $\mathcal{E}$ to extract a feature when the sample is attached to a binary domain index d to indicate its origin. Given a feature, $\mathcal{C}$ calculates the prediction of the current stage y^{pred}. Then, the staging loss L^c is computed using a cross-entropy function as in [31].

$\mathcal{D}$ outputs a set of domain prediction $\{d^{dom}, d^{stg}, d^{sub}\}$ at the same time in the forward pass. The discrimination loss is defined as the sum of the losses described below:

$$L^d = L^{dom} + L^{stg} + L^{sub}, \tag{1}$$

where L^{dom} is a global discrimination loss that is calculated from the cross-entropy loss function. It measures the difference between d and d^{dom}. L^{stg} and L^{sub} are similarly computed to fix possible local misalignments left from using only L^{dom}. More details about the loss functions are given in Section IV.

In the backward pass, the network is updated to minimize the classification loss as below:

$$\min_{\mathcal{C},\mathcal{E}} L^c, \tag{2}$$

where the network attempts to conduct the adaptation through a minimax game on the discrimination loss between $\mathcal{D}$ and $\mathcal{E}$. The optimization procedure is given as below:

$$\min_{\mathcal{D}} \max_{\mathcal{E}} L^d. \tag{3}$$

Specifically, $\mathcal{C}$ is updated first to minimize L^c by computing a gradient by conventional backpropagation. $\mathcal{D}$ is also updated to minimize L^d. Then, the gradient $\frac{\partial L^d}{\partial \mathcal{D}}$ is reversed for adversarial learning for $\mathcal{E}$ [26], [33]. Then, the gradients from $\mathcal{C}$ and $\mathcal{D}$ interflow for $\mathcal{E}$. Therefore, $\mathcal{E}$ is updated by minimizing L^c and maximizing L^d simultaneously. While a classification loss decreases and a discrimination loss increases through the adversarial learning, the knowledge is transferred in a way that $\mathcal{D}$ is deceived in the aligned feature space and $\mathcal{E}$ produces a domain-invariant feature.

More thorough explanations about the optimization and the loss functions will be given in Section IV.

2) Inference: The workflow differs for the training and test phases. In the inference phase, a sample in $\{x_j^t\}_{j=1}^{m_{te}}$ from the target is tested to produce a predicted probability of the current sleep stage as follows:

$$\hat{y}^{pred} = \mathcal{C}(\mathcal{E}(x_j^t)).$$

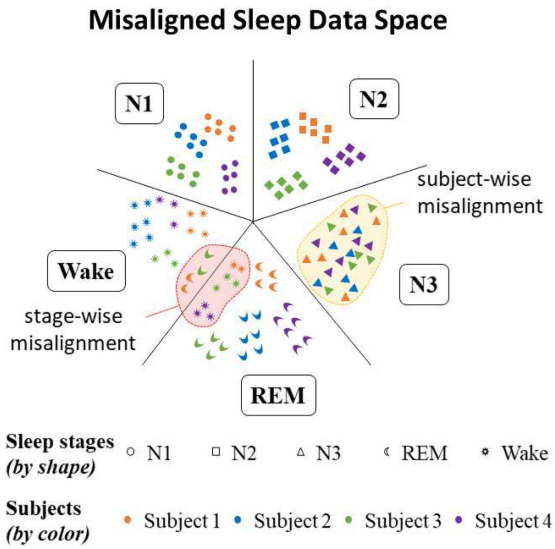

Fig. 2. Local misalignments of sleep data.

B. Discriminators for Domain Alignments

The proposed method attempts to improve sleep staging performance in the target. This objective is accomplished by exploiting both the global and local alignments in the proposed method. Global adaptation plays a role in training a domain-adapted model across different datasets and has been actively used in previous works [26], [30], [33]. However, local adaptation has been overlooked thus far although the models fail to decide upon accurate classification boundaries. In the proposed method, we define a common structure of the sleep data to consider local alignments across two domains. Specifically, we develop a stage discriminator and a subject discriminator. These discriminators support preservation of the inherent structure of a sleep sample and improve performance after DA. Fig. 2 illustrates the distribution of sleep data and importance of local attributions. Although the global distribution is aligned, local distributions are still misaligned as in the image.

Subject-wise alignment is considered when the distribution of a sleep sample varies from person to person. It includes an existence of a sleep disorder, an age, and behavioral or physiological conditions [40], [41]. For instance, the sensitivity of a monitoring device can be different between a PSG in a hospital and a wearable device. In our experiments, we differentiate the characteristics of the source and target with different acquisition and environmental conditions, e.g., MASS dataset [23] as source and Sleep-EDF [42], [43] as target. In addition, we use sleep data samples from normal subjects as the source but some data from patients with sleep difficulty as the target, e.g., MASS dataset as source and Sleep-EDF-st dataset [42], [43] as target.

On the other hand, stage-wise alignment considers the inherent characteristics of sleep signals. For example, brain neural activity during REM sleep is comparable to when awake [24],

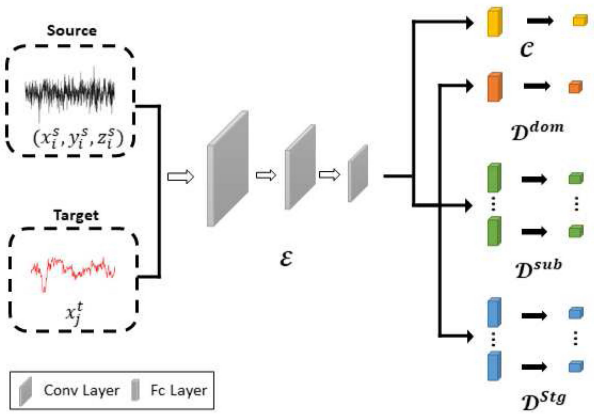

Fig. 3. Implementation of a network architecture.

so the distribution at the REM stage is close to that at the wake stage. Taking Fig. 2 as an example, the "W" stage of subjects 1, 3, and 4 can be aligned to the REM space. The stage-wise alignment can also reflect external factors, such as an electrical noise.

We provide more explanation about the discriminators. $\mathcal{D}^{dom}$, which is a global domain discriminator, determines if a feature f_i is extracted from the source or target. The discriminator aims to provide domain-invariant representation across $\mathbf{X}^S$ and $\mathbf{X}^T$. $\mathcal{D}^{stg}$ is a stage discriminator to resolve discrepancies between the same stage in the other domain. Specifically, the k^{th} stage of a source domain is aligned to the same stage of a target domain using $\mathcal{D}_k^{stg}, k \in \mathbf{C}^S$. $\mathcal{D}_l^{sub}$ is a subject discriminator for the l^{th} subject and is designated for achieving subject-wise adaptation. In the proposed method, the local discriminators are incorporated to resolve local distribution misalignments and efficiently transfer knowledge to a target domain.

C. Network Implementation

We show more detailed implementations of the network. In Fig. 3, the discriminators depicted as orange, green, and blue bars are composed of the branches of multiple fully connected layers to output a logit for domain prediction. The global discriminator is composed of a single branch, while the local discriminators have multiple branches. The number of branches for the local discriminators depends on the number of stages and subjects that constitute the domain. Specifically, there are five branches for the stage discriminator reflecting the number of sleep stages. The number of branches for the subject discriminator differs by the number of individuals or their groups in the source domain. In the experiment, we designed five subject discriminators based on the number of subsets from the database used as the source domain.

IV. Training

It is not a trivial problem to train the proposed sleep staging network because the conventional adversarial learning approach is not applicable to our problem. We present several loss functions for training and optimizations in this section.

A. Loss Function

1) Classification Loss Function: We use a classification loss function to perform sleep staging. The overall classification loss function is given as

$$\begin{aligned} L^c &= L_c^s + \alpha L_c^t \\ &= \sum_{x_i^s \in \mathbf{X}^S} H(y_i^{pred}, y_i^s) + \alpha \sum_{x_j^t \in \mathbf{X}^T} \hat{H}(y_j^{pred}), \end{aligned} \tag{4}$$

where L_c^s is the cross-entropy loss function for a source sample. x_i^s and y_i^s are an input sample and the corresponding label, respectively. $H(\cdot,\cdot)$ represents the cross-entropy loss function, and y_i^{pred} is the softmax output of $\mathcal{C}$. L_c^t is the conditional cross-entropy loss function for a target sample [31], [35], computed by measuring a confidence to the prediction itself due to the lack of a label. $\hat{H}(p_j^t) = -\sum_{k=1}^{r} p_j^t(k) \log p_j^t(k)$. $p_j^t(k)$ is the k^{th} element of p_j^t, which is the probability that x_j^t belongs to the k^{th} class. α is an adaptive hyper parameter to balance the staging loss in the target. We will explain how α is determined later.

2) Discrimination Loss Functions: The discrimination loss in (1) is defined as the sum of the domain discrimination loss function L^{dom}, the subject-discrimination loss function L_l^{sub} used for a sample belonging to the l^{th} subject, and the stage-discrimination loss function L_k^{stg} used for a sample belonging to the k^{th} stage, given as

$$\begin{aligned} L^d &= L^{dom} + L^{stg} + L^{sub} \\ &= L^{dom} + \sum_{l=1}^{b} L_l^{sub} + \sum_{k=1}^{r} L_k^{stg}, \end{aligned} \tag{5}$$

where b and r are the number of subjects and stages, respectively.

L^{dom} is defined as

$$L^{dom} = \sum_{x_i \in \mathbf{X}^S \cup \mathbf{X}^T} H(d_i^{dom}, d_i), \tag{6}$$

where d_i^{dom} is an output from $\mathcal{D}^{dom}(\mathcal{E}(x_i))$, and d_i is set to 1 if $x_i \in \mathbf{X}^S$ and 0 otherwise.

L_k^{stg} is defined as

$$L_k^{stg} = \sum_{x_i \in \mathbf{X}^S \cup \mathbf{X}^T} p_i^k H(d_i^{stg_k}, d_i), \tag{7}$$

where $d_i^{stg_k}$ is an output from $\mathcal{D}_k^{stg}(\mathcal{E}(x_i))$ and $k \in \{1, 2, \ldots, 5\}$ is an index of the sleep stage. p_i^k is the probability that x_i belongs to the k^{th} class. For $x_i \in \mathbf{X}^S$, p_i^k is set to 1 if y_i is equal to k and 0 otherwise. For $x_i \in \mathbf{X}^T$, p_i^k is computed from the output logit of $\mathcal{C}$ owing to the lack of a label.

L_l^{sub} is defined as

$$L_l^{sub} = \sum_{x_i \in \mathbf{X}^S \cup \mathbf{X}^T} p_i^l H(d_i^{sub_l}, d_i), \tag{8}$$

where $d_i^{sub_l}$ is an output from $\mathcal{D}_l^{sub}(\mathcal{E}(x_i))$, $l \in \{1, 2, \ldots, b\}$ is an index of the l^{th} subject discriminator, and p_i^l is the probability that x_i belongs to the l^{th} subject. The subject is revealed with a label z_i along with $x_i \in \mathbf{X}^S$. p_i^l is set to 1 if $x_i \in \mathbf{X}^S$ belongs to the l^{th} subject (i.e., $z_i = l$) and 0 otherwise. For $x_i \in \mathbf{X}^T$, p_i^l is set to $\frac{1}{b}$. The number of parameters in D_l^{sub} increases with the

number of subjects. This can be a burden during training, so we assign a discriminator to a group of subjects instead of all the individuals.

3) Gradient-Synchronization Loss Function: The training converges slow if each gradient computed from individual discriminators exhibits a different direction. Therefore, we use an optimization scheme to make both the direction and the slope of each gradient be consistent. Specifically, we use a gradient-synchronization loss to align gradients of global and local discriminators as a regularization, motivated by [35].

$L^{syn}_{sub\sim stg}$ is used to align the gradients for learning a stage-discriminator and a subject-discriminator given as

$$L^{syn}_{sub\sim stg} = \left| \sum\nolimits_{x_i \in \mathbf{X}^S \cup \mathbf{X}^T} \left\| \frac{\partial L^{sub}_l}{\partial \mathcal{E}(x_i)} \right\|_2 - \sum\nolimits_{k \in sub_l} \sum\nolimits_{x_i \in \mathbf{X}^S \cup \mathbf{X}^T} \left\| \frac{\partial L^{stg}_k}{\partial \mathcal{E}(x_i)} \right\|_2 \right|, \quad (9)$$

where the first term computes an amount of the gradient in the l^{th} subject-discriminator, and the second term computes the sum of the gradients of all the stage-discriminators of the l^{th} subjects.

$L^{syn}_{dom\sim sub}$ is used to align the gradients of a domain-discriminator and a subject-discriminator, given as

$$L^{syn}_{dom\sim sub} = \left| \sum\nolimits_{x_i \in \mathbf{X}^S \cup \mathbf{X}^T} \left\| \frac{\partial L^{dom}}{\partial \mathcal{E}(x_i))} \right\|_2 - \sum\nolimits_{l=1}^{b} \sum\nolimits_{x_i \in \mathbf{X}^S \cup \mathbf{X}^T} \left\| \frac{\partial L^{sub}_l}{\partial \mathcal{E}(x_i)} \right\|_2 \right|, \quad (10)$$

where the first term is the sum of gradients of the domain discriminator, and the second term is the sum of gradients for all subject discriminators.

After all, the gradient-synchronization loss is given as

$$L^{syn} = \frac{1}{b} \sum_{l=1}^{b} L^{syn}_{sub\sim stg} + L^{syn}_{dom\sim sub}. \quad (11)$$

The direction and slope of all gradient descents can be synchronized using this regularizing term.

B. Overall Objective

We have defined the objective functions in (2) and (3). The overall optimization process for each step is explained to show the updates of learning parameters, denoted as θ with subscripts indicating their relevant modules.

First, the stage classifier $\mathcal{C}$ is optimized to minimize the classification loss in (2) with a learning rate η

$$\theta_{\mathcal{C}} \leftarrow \theta_{\mathcal{C}} - \eta \frac{\partial L^c}{\partial \theta_{\mathcal{C}}}. \quad (12)$$

When $\mathcal{E}$ is given, the discriminators $\mathcal{D} = \{\mathcal{D}^{dom}, \mathcal{D}^{sub}, \mathcal{D}^{stg}\}$ are optimized by minimizing the loss combining the discrimination loss in (1) and the synchronization loss in (11) as below:

$$\min_{\mathcal{D}} L^d + \beta L^{syn}, \quad (13)$$

where β is a hyperparameter. It is shown later how β is determined.

Then, relevant parameters of the discriminators are updated as follows:

$$\theta_{\mathcal{D}^{dom}} \leftarrow \theta_{\mathcal{D}^{dom}} - \eta \frac{\partial (L^{dom} + \beta L^{syn})}{\partial \theta_{\mathcal{D}^{dom}}},$$
$$\theta_{\mathcal{D}^{sub}} \leftarrow \theta_{\mathcal{D}^{sub}} - \eta \frac{\partial (L^{sub} + \beta L^{syn})}{\partial \theta_{\mathcal{D}^{sub}}},$$
$$\theta_{\mathcal{D}^{stg}} \leftarrow \theta_{\mathcal{D}^{stg}} - \eta \frac{\partial (L^{stg} + \beta L^{syn})}{\partial \theta_{\mathcal{D}^{stg}}}. \quad (14)$$

On the other hand, when $\mathcal{D}$ is given, $\mathcal{E}$ are optimized to make a feature to be discriminative and domain invariant at the same time. In order to train the domain-invariant feature, the gradients of the discriminators are reversed by the gradient reversal layer [26], [33], thus allowing the minimax game between the stage classifier and the discriminators defined in (3). Mathematically, the gradient reversal is performed in by taking negative of the discrimination losses. This can be achieved by minimizing the classification loss and simultaneously confusing the discriminators as below:

$$\min_{\mathcal{E}} L^c + \beta L^{syn} - L^d. \quad (15)$$

Then its parameters are updated as follows:

$$\theta_{\mathcal{E}} \leftarrow \theta_{\mathcal{E}} - \eta \left(\frac{\partial (L^c + \beta L^{syn})}{\partial \theta_{\mathcal{C}}} \cdot \frac{\partial \theta_{\mathcal{C}}}{\partial \theta_{\mathcal{E}}} - \frac{\partial L^d}{\partial \theta_{\mathcal{D}}} \cdot \frac{\partial \theta_{\mathcal{D}}}{\partial \theta_{\mathcal{E}}} \right). \quad (16)$$

C. Adaptive Hyper Parameter Tuning

In (4) and (13), α and β are used as weighting factors of different loss functions. We observe that the network tends to be overfitted to the source when the hyperparameters are inappropriately determined. This problem occurs because the conditional cross-entropy loss cannot reduce an error during backpropagation owing to the lack of labels. To avoid a trial and an error manner, we adjust the parameters dynamically during training. We examine the ratio between a source loss and a target loss for each training epoch and update the parameters as follows:

$$\alpha \leftarrow \frac{L^s_c}{L^t_c} \cdot \frac{1}{p} \qquad \beta \leftarrow \frac{L^{dis}}{L^{syn}} \cdot \frac{1}{p}, \quad (17)$$

where L^s_c and L^t_c are the cross-entropy losses in the source and target, respectively. $L^{dis} = \text{mean}(L^{dom}, L^{sub}, L^{stg})$ is the arithmetic mean between all discrimination losses. p is a constant parameter used to adjust the ratio.

The ratios of the loss terms usually fluctuate rapidly in the early stage of training, so it would be inefficient to update the hyperparameters during every iteration. Instead, we update the hyperparameters once every K epochs. Specifically, we compute the average of the ratios during K epochs and update the parameters. We show these procedures in Algorithm 1 with a maximum training epoch denoted as L.

Algorithm 1: Adaptive Hyperparamter Tuning.

alphaRatioList =[], betaRatioList = []
for $i = 1, \ldots, L$ **do**
  Add $\frac{L_c^s}{L_c^t}$ to alphaRatioList
  Add $\frac{L^{dis}}{L^{syn}}$ to betaRatioList
  Update network weights $\theta_{\mathcal{C}}, \theta_{\mathcal{D}}, \theta_{\mathcal{E}}$ by (12), (14) and (16)
  if i *mod* K=0 **then**
    $\frac{L_c^s}{L_c^t} \leftarrow$ mean(alphaRatioList)
    $\frac{L^{dis}}{L^{syn}} \leftarrow$ mean(betaRatioList)
    Update hyperparamters α, β by (17)
    Empty alphaRatioList and betaRatioList
  end
end

V. Experiments

A. Experimental Settings

1) Source-Domain Database: We use the Montreal Archive of Sleep Studies (MASS) [23] database as the source domain because this dataset contains a large number of biometric signals obtained from different sleep laboratories. MASS consists of all-night sleep data for 97 men and 103 women whose ages range from 18 to 76 years. It is known that the subjects are all healthy. Subject-specific demographic features are restricted to access. The sleep data were constructed manually by several sleep experts, following the AASM standards [5] to annotate the SS1 and SS3 subsets and the R&K standards [4] to annotate the other stages. As in [10], the annotations are converted to W, N1, N2, N3, and REM by combining N3 and N4 in the N3 stage and neglecting MOVEMENT and UNKNOWN. The 20 s epochs were extended to 30 s ones by padding each 5 s segment of temporally neighboring epochs.

2) Target-Domain Database: We use the Sleep-EDF database [42], [43] and Sleep-EDF-st database [42], [43] for the target domain. The numbers of subjects were 20 and 22, respectively. We apply the leave-one-out cross-validation (CV) for Sleep-EDF and 11-fold CV for Sleep-EDF-st as in [10], [27]. The recordings are trimmed from light-off time to light-on time both in the databases as in [27]. The main difference in the data is the health conditions of subjects. We show more details about the databases below.

Sleep-EDF database includes 20 subjects ranging from 25 to 34 years. The PSG data were recorded over two nights, except for one subject whose data is for only one night. Each epoch has eight categories {W, N1, N2, N3, N4, REM, MOVEMENT, UNKNOWN}, in accordance with the R&K standard [4]. As in [10], we merged the N3 and N4 stages into N3; the MOVEMENT and UNKNOWN categories were excluded in the experiments.

Sleep-EDF-st database is a subset of the Sleep-EDF expanded database [42], [43], including 22 Caucasian subjects ranging from 18 to 79 years. It is known that the subjects had mild difficulty falling asleep.

TABLE I
Input Channels of PSG Montages

Database	Subject	EEG	EOG	EMG
MASS	200	C4-A1	ROC-LOC	CHIN1-CHIN2
Sleep-EDF	20	Fpz-Cz	ROC-LOC	-
Sleep-EDF-st	22	Fpz-Cz	ROC-LOC	Submental

3) PSG Montages: We conducted experiments using various PSG montages. The test performance is presented with using the single channel data (i.e. EEG $\rightarrow$ EEG and EOG $\rightarrow$ EOG) and the multi-channel data (i.e. EEG $\cdot$ EOG$\cdot$EMG $\rightarrow$ EEG$\cdot$ EOG$\cdot$EMG, EEG $\cdot$ EOG$\rightarrow$ EEG $\cdot$ EOG). We summarized the input channels in Table I. EMG recordings are not available in the Sleep-EDF database.

4) Sleep-Staging Network: We used various baseline networks such as DeepSleepNet [10] and SeqSleepNet [11] to verify the effectiveness of the proposed method.

- **DeepSleepNet** [10] takes a single epoch for an input sleep data. A feature extractor provides time-independent features in a single epoch. The original training scheme explained in [10] is applied.
- **SeqSleepNet** [11] takes a spectrogram as an input and uses a multi-sequence classification to handle multiple epochs and produce corresponding results from the epochs.
- **DeepSleepNet+** and **SeqSleepNet+** are improved versions of DeepSleepNet and SeqSleepNet, respectively, in [27]. They also conduct multi-sequence classification.

The proposed method was deemed applicable to the models because it is largely model agnostic. In our implementations, for all the discriminators, we use two fully connected layers with 512 neurons and 64 neurons for DeepSleepNet and SeqSleepnet, respectively. We use a softmax layer to determine whether an input is from the source or target domain. When the proposed method is applied to a multi-sequence classification framework, it produces the same number of domain predictions as the input epochs. In the original SeqSleepNet [11], DeepSleepNet+ and SeqSleepNet+ [27], it handles 20 epochs for a prediction result. However, in our implementation, we reduced the number to 5 because the models require additional parameters for domain discriminators. The models are retrained due to these changes.

The networks are trained with one global domain discriminator and five stage discriminators. We use a subject discriminator used for each of subject groups. We use five subject discriminators in the implementation because there are five subject groups in the MASS database, in which the subject groups are configured with different acquisition devices, filtering methods, etc. The configurations can reflect subject-wise domain discrepancies.

5) Metrics: The performance is evaluated using overall accuracy, a macro F1-score, and Cohen's kappa (κ) as the metrics, calculated as below:

$$\text{ACC} = \frac{\sum_{c \in \mathbf{C}^S} \text{TP}_c}{N} \tag{18}$$

$$\text{MF1} = \frac{\sum_{c \in \mathbf{C}^S} \text{F1}_c}{5}, \tag{19}$$

TABLE II
STAGING PERFORMANCE ON THE SOURCE DOMAIN

Input	DeepSleepNet			SeqSleepNet		
	Acc.	MF1	κ	Acc.	MF1	κ
EEG·EOG·EMG	85.5	83.1	0.790	86.5	82.9	0.799
EEG·EOG	84.8	81.0	0.783	85.6	81.4	0.793
EEG	84.3	80.7	0.780	84.4	80.0	0.774
EOG	81.9	78.0	0.745	82.5	76.7	0.746

where TP_c and F1_c are the true positive and per-class F1 score of class $c \in \mathbf{C}^S$, respectively, and N is the total number of test samples.

κ [44] is a statistical measure of the interrater agreement (IRA) level calculated as

$$\kappa = \frac{\sum_{c\in\mathbf{C}^S} p_{cc} - \sum_{c\in\mathbf{C}^S} p_{c+}p_{+c}}{1 - \sum_{c\in\mathbf{C}^S} p_{c+}p_{+c}} = \frac{p_a - p_e}{1 - p_e}, \tag{20}$$

where p_{cc} represents the percentage of epochs classified as category c by the network and the annotated label simultaneously, and p_{c+} and p_{+c} represent the percentages of epochs classified as category c by the network and annotated label, respectively.

B. Experimental Results

1) Performance on Source Domain: Table II presents the test staging performance of [10] and [11] on the MASS database as source domain. The models are trained from scratch, using 180 subjects for training, 10 subjects for validation, and 10 subjects for testing in the MASS database. The previous works have reported the similar performance [11], [27], and the results verify the applicability of the proposed method in different PSG montages. In the results, SeqSleepNet slightly outperforms DeepSleepNet in various input PSG montages, by taking an advantage of a multi-sequence prediction.

2) Performance on Target Domain: Table III presents the test staging performance on target domain. We configure compared learning schemes or conditions to evaluate the performance of the proposed method as follows.

Scratch learning (SC) refers to a scenario under a condition that a network is trained and tested using disjoint samples belonging to the same target domain. It does not use a transfer learning but needs ground truth target data during training. A network can exploit the training and testing samples chosen from the same database.

Direct transfer (DT) refers to a scenario under that a network is trained in the source and straightforwardly tested in the target. This condition would be adversarial to a network owing to different characteristics of the training and testing samples.

Fine tuning (FT) uses labeled training samples obtained from both source and target domains so can exploit a larger number of samples during transfer learning. It cannot perform the transfer learning properly if there are no ground truth data in target domain, which falls back to DT. We present the results of [27] as the state-of-the-art supervision.

The proposed method is compared with these three settings as shown in Table III. "CD" and "US" describe cross-domain knowledge transfer and unsupervised learning settings. In our work, the performance is expected to be close to those in the SC and FT and higher than DT although the ground truth is unknown during the training. The number of samples in the source domain is larger than that in the target domain. Thus, we use the training and testing samples differently. In Table III, the specific numbers of subjects are denoted by N_s and N_t, used for training in source domain ("S") and target domain ("T"), respectively. N_t^u represents the unlabeled training data. N_s is 200 for the FT as in [27]. For the other configurations, N_s is 180. N_t is 19 and 20 for Sleep-EDF and Sleep-EDF-st, respectively. N_t^u is equal to N_t. We use the same test sets for fair comparisons.

In the Sleep-EDF dataset, the performances of DT is degraded 6–17.1% (10.8% on average) and 3.4–14.4% (8.1% on average) in overall accuracy (Acc.) compared to those of SC for DeepSleepNet and SeqSleepNet. In the Sleep-EDF-st dataset, the performance drop reduces to 0.3–8.3% (3.2% on average) and 2.2–12.8% (4.6% on average) in overall accuracy for DeepSleepNet and SeqSleepNet, respectively. In the results, SC provides a superior performance to DT in various settings. In fact, SC presents the best expectation of a network because a network can exploit the training and testing samples chosen from the same database [45]. In contrast, DT would be adversarial to a network owing to different characteristics of the training and testing samples. Some results such as a single EEG channel of Sleep-EDF-st dataset display improved performance of DT over SC. We think the small number of training samples would affect the performance. N_t is significantly smaller than N_s as shown in Table III.

The proposed method attempts to reduce the domain discrepancies and efficiently recovers the performance drop of DT even though the knowledge transfer is conducted using unlabeled samples in target domain. For DeepSleepNet, as compared to DT, the performances of the proposed method are improved approximately 6.5%, 8.8%, and 0.08 in terms of Acc., MF1, and κ in the Sleep-EDF dataset. We observe similar results in SeqSleepNet, in which the improved performances are 2.8%, 1.8%, and 0.04 on average. The performance difference decreases because the performance differences between DT and SC are smaller in SeqSleepNet. For DeepSleepNet+ and SeqSleepNet+, the accuracies of the proposed method are improved approximately 7.4% and 1.9%, respectively, as compared to DT. We also examine the results in the Sleep-EDF-st dataset. For DeepSleepNet and SeqSleepNet, the accuracies of the proposed method are improved approximately 4.6% and 4.7%, respectively, as compared to DT. We observe similar behaviors in the other baseline models. In the accuracies, we observe that the best record is 10.4% improvement as compared to DT, which is applied to a single EOG channel data of Sleep-EDF using DeepSleepNet+. We observe the proposed method enhances the performance over DT in almost all of the cases, but, for a single EOG channel data of Sleep-EDF using SeqSleepNet+, there is a slight loss. However, it is noted that the number of training samples in DT are significantly larger than that of the proposed method.

In FT [27], the baseline models are finetuned to improve performance in target domain, by using labeled samples as many as $N_s + N_t$. DeepSleepNet+ and SeqSleepNet+ are designed for transfer learning, so they achieve the best performance among

TABLE III
QUANTITATIVE RESULTS OF THE PROPOSED DOMAIN ADAPTATION (PROP) USING VARIOUS METRICS AS COMPARED WITH SC, DT, AND FT [27] AS THE STATE-OF-THE-ART SUPERVISION. "CD" AND "US" REFER TO CROSS-DOMAIN KNOWLEDGE TRANSFER AND UNSUPERVISED LEARNING SETTINGS, RESPECTIVELY

	Methods	Model	CD/US	Number of training data		EEG·EOG·EMG →EEG·EOG·EMG			EEG·EOG →EEG·EOG			EEG→EEG			EOG→EOG		
				S	T	Acc.	MF1	κ	Acc.	MF1	κ	Acc.	MF1	κ	Acc.	MF1	κ
Sleep-EDF	SC	DeepSleepNet	No/No	-	N_t	-	-	-	80.7	75.2	0.740	79.5	74.3	0.733	77.9	70.1	0.686
		DeepSleepNet+	No/No	-	N_t	-	-	-	81.7	75.2	0.742	80.7	74.2	0.730	75.9	66.8	0.651
		SeqSleepNet	No/No	-	N_t	-	-	-	82.2	73.7	0.744	82.0	72.3	0.743	78.8	67.5	0.692
		SeqSleepNet+	No/No	-	N_t	-	-	-	82.2	74.2	0.744	82.2	74.1	0.746	78.5	68.3	0.688
	DT	DeepSleepNet	Yes/No	N_s	-	-	-	-	71.5	63.2	0.611	73.5	66.1	0.636	60.8	47.2	0.449
		DeepSleepNet+	Yes/No	N_s	-	-	-	-	70.3	59.7	0.586	74.0	66.9	0.648	57.1	44.9	0.426
		SeqSleepNet	Yes/No	N_s	-	-	-	-	75.7	67.7	0.652	78.6	69.7	0.697	64.4	52.7	0.478
		SeqSleepNet+	Yes/No	N_s	-	-	-	-	72.0	62.1	0.600	81.2	74.6	0.733	67.2	59.1	0.530
	FT [27]	DeepSleepNet	Yes/No	N_s	N_t	-	-	-	77.9	71.0	0.683	79.9	74.8	0.722	79.5	72.9	0.722
		DeepSleepNet+	Yes/No	N_s	N_t	-	-	-	84.4	78.9	0.782	84.4	78.8	0.779	79.6	73.2	0.713
		SeqSleepNet	Yes/No	N_s	N_t	-	-		81.4	75.6	0.755	83.4	76.2	0.765	79.0	70.9	0.700
		SeqSleepNet+	Yes/No	N_s	N_t	-	-	-	84.2	77.5	0.774	85.1	79.6	0.787	81.7	75.0	0.736
	PROP	DeepSleepNet	Yes/Yes	N_t	N_t^u	-	-	-	74.9	68.8	0.654	80.7	73.4	0.737	69.7	60.8	0.549
		DeepSleepNet+	Yes/Yes	N_t	N_t^u	-	-	-	75.7	65.4	0.637	80.3	70.1	0.728	67.5	55.0	0.527
		SeqSleepNet	Yes/Yes	N_t	N_t^u	-	-	-	79.6	69.7	0.705	79.9	69.8	0.708	67.5	56.0	0.520
		SeqSleepNet+	Yes/Yes	N_t	N_t^u	-	-	-	77.6	66.8	0.679	81.4	70.6	0.733	67.1	56.6	0.515
Sleep-EDF-st	SC	DeepSleepNet	No/No	-	N_t	76.7	71.8	0.664	75.1	70.2	0.641	73.1	65.9	0.605	67.3	56.9	0.509
		DeepSleepNet+	No/No	-	N_t	73.7	69.6	0.631	73.5	67.5	0.623	72.4	64.4	0.601	69.9	65.9	0.573
		SeqSleepNet	No/No	-	N_t	78.9	72.3	0.700	79.1	72.0	0.703	76.8	65.8	0.668	78.8	69.8	0.695
		SeqSleepNet+	No/No	-	N_t	79.4	74.5	0.707	79.4	74.8	0.709	76.4	70.5	0.667	78.6	71.6	0.692
	DT	DeepSleepNet	Yes/No	N_s	-	72.7	66.3	0.620	73.9	70.0	0.59	64.9	53.9	0.51	67.0	57.1	0.526
		DeepSleepNet+	Yes/No	N_s	-	74.4	67.4	0.644	71.6	65.3	0.610	66.7	61.3	0.542	70.0	63.2	0.584
		SeqSleepNet	Yes/No	N_s	-	76.7	68.9	0.663	75.9	68.0	0.655	78.9	72.4	0.700	66.0	55.6	0.496
		SeqSleepNet+	Yes/No	N_s	-	79.2	73.1	0.701	73.1	64.3	0.614	80.5	75.6	0.719	67.2	59.2	0.529
	FT [27]	DeepSleepNet	Yes/No	N_s	N_t	79.0	74.2	0.690	79.7	74.5	0.703	77.8	73.5	0.679	77.0	71.3	0.657
		DeepSleepNet+	Yes/No	N_s	N_t	80.1	76.2	0.719	80.1	75.8	0.719	81.5	77.4	0.729	77.3	74.1	0.681
		SeqSleepNet	Yes/No	N_s	N_t	82.1	77.9	0.746	80.8	77.4	0.736	81.1	77.2	0.743	80.1	75.4	0.727
		SeqSleepNet+	Yes/No	N_s	N_t	80.5	76.1	0.725	80.9	76.8	0.734	81.0	77.5	0.737	80.3	76.1	0.722
	PROP	DeepSleepNet	Yes/Yes	N_t	N_t^u	75.6	69.8	0.614	75.2	68.6	0.637	74.6	68.0	0.631	69.9	61.2	0.552
		DeepSleepNet+	Yes/Yes	N_t	N_t^u	74.5	68.1	0.638	72.8	66.2	0.628	71.7	63.5	0.615	68.1	60.8	0.558
		SeqSleepNet	Yes/Yes	N_t	N_t^u	79.7	70.0	0.708	79.7	67.8	0.705	78.4	65.6	0.686	74.8	64.3	0.634
		SeqSleepNet+	Yes/Yes	N_t	N_t^u	78.1	71.1	0.700	79.3	70.4	0.701	77.1	65.1	0.671	73.3	62.5	0.613

compared baseline models. To be specific, for DeepSleepNet+, as compared to DT, the performances of the FT are improved approximately 15.7%, 19.8%, and 0.21 in terms of Acc., MF1, and κ in the Sleep-EDF dataset. For SeqSleepNet+, the performance improvements are 10.2%, 12.3%, and 0.15 on average. On contrary, DeepSleepNet and SeqSleepNet do not take an advantage of fine-tuning as much. Although the performance of the FT is largely better than the proposed method, the FT needs labeled samples in target domain. However, they are usually unavailable in the real world whereas the proposed method relaxes the assumption. The performance of the FT drops to DT when there is no labeled target samples. Furthermore, the number of training samples in source domain is mattered to provide reliable performance in the FT. In comparisons, the proposed method uses N_t labeled samples. We exhibit more results on the performance change of the FT with N_s in Section V-C5.

We show more comprehensive results for the proposed method, the SC, and the DT in Fig. 4. Each graph represents the values of overall accuracy, MF1 scores, Cohen's kappa, precision, and recall values of the tested models. The precision and recall values are computed as the means of the per-class precisions and recalls. It is clearly seen in DeepSleepNet that the performance of the proposed algorithm approximates to that of the SC. In SeqSleepNet, there are small margins between the SC and DT. Nevertheless, the performance of the proposed algorithm is close to that of the SC.

Moreover, the performance of the proposed method is compared with the state-of-the-art unsupervised DA scheme [39]

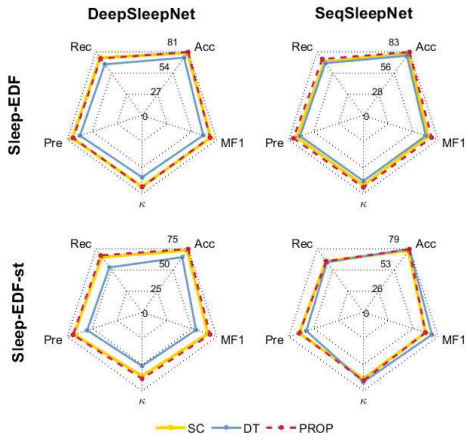

Fig. 4. Illustration of the quantitative results in the different baseline models and datasets. Best viewed in color.

for sleep-EDF as target. [39] uses multiple channel discriminators for input data channels, so we adopt two channel-wise discriminators for EEG · EOG and one discriminator for an EEG input. Other training schemes were performed same as described in [39]. Table IV displays the enhanced performance (Δ) of the proposed method as compared to [39] in the same training conditions. N_t and N_t^u are the same, too. The proposed method outperforms [39] approximately 4.8%, 3.9%, and 0.05 on average in terms of Acc., MF1, and κ for multi-channel EEG · EOG input data and 8.5%, 7.8% and 0.11 for a single EEG

TABLE IV
PERFORMANCE COMPARISONS WITH [39]. THE ENHANCED PERFORMANCE OF THE PROPOSED METHOD IS REMARKED WITH Δ

Method	Model	EEG·EOG →EEG·EOG			EEG → EEG		
		Acc.	MF1	κ	Acc.	MF1	κ
Nasiri [39]	DeepSleepNet	Δ4.2	Δ4.9	Δ0.052	Δ9.8	Δ10.0	Δ0.119
	DeepSleepNet+	Δ4.1	Δ1.2	Δ0.035	Δ7.8	Δ6.3	Δ0.103
	SeqSleepNet	Δ3.9	Δ4.5	Δ0.054	Δ5.5	Δ5.7	Δ0.074
	SeqSleepNet+	Δ6.9	Δ4.9	Δ0.074	Δ10.7	Δ9.3	Δ0.126

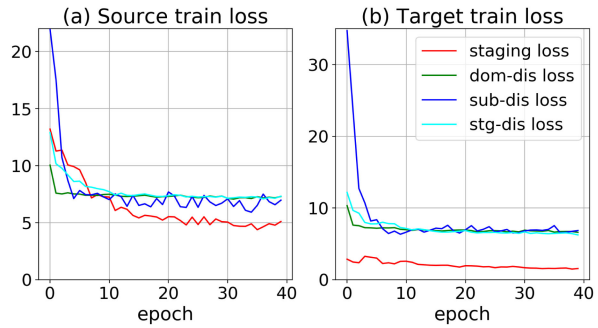

Fig. 5. Training curves of a staging loss function and three discrimination loss functions for (a) source samples and (b) target samples.

input data. The results demonstrate that the proposed method exploits better global and local domain discriminators to reflect the intrinsic structures of sleep data.

C. Discussion

In this subsections, all results are presented using DeepSleepNet and a single EEG channel data of the Sleep-EDF.

1) Adversarial Training Analysis: We attempt to justify the performance of the results by showing the intermediate training results. Fig. 5 shows the loss curves of the DeepSleepNet for validation of fold 0 during training. Practically, in our training, we first use source samples and then input the remaining samples from the target. Fig. 5a and 5b are plotted when training the source and target samples, respectively. Losses depicted in Fig. 5 represent the means of losses calculated from all batches at each training epoch.

Our model is trained with reducing the staging losses both in the source and target. The staging loss in source decreases substantially because the parameters are updated through standard backpropagation. The staging loss in the target is relatively stable because it is computed from the confidence of model prediction using conditional cross-entropy in (4). It is noted that the three losses of a domain discriminator, subject discriminator, and stage discriminator decrease in the early learning time, while the discrimination losses remain constant after a certain point. This phenomenon shows that a feature learns the dominant-invariant property by fooling the discriminators. Therefore, we find that the classifier and discriminators in the proposed method are trained suitably.

2) Performance Analysis of Adversarial Discriminators: We investigate the performance changes by turning on or off the local discriminators to verify the effectiveness of the discriminators. Table V(a) shows the specific conditions for comparisons, using the Sleep-EDF dataset as the target domain and DeepSleepNet as the base model. The best performance is 80.7% when all the discriminators are applied. However, the performance degrades to 77.8% and 75.4% when disabling the stage- and subject-discriminators, respectively. In particular, the subject discriminator gives more impact to the performance. It is noticed that the performance in the DT was about 73.5%. The domain-discriminator enhances the performance from 73.5% to 75.3%. In addition, the performance is improved with the local discriminators by fixing the misaligned local attributes.

Fig. 6 shows the results of confusion matrices when disabling the local discriminators. From the left to the right, we display the results of the DT, "Dom," using only the domain discriminator, "Dom+Sub+Stg," using all the domain discriminators, and SC. We observe that the diagonal elements of the matrices exhibit increasing performance from the left to the right. "Dom" in Fig. 6 shows improved performance to DT. However, it is noted that the performance for N1 and N3 stages is rather poor. These results show that the overall distribution is aligned but there are still misalignments in the stage distributions. On the other hand, "Dom+Sub+Stg" in Fig. 6c presents the similar performance to that in the SC in Fig. 6d.

3) Performance Analysis With the Number of Subject Discriminators: We examine the performance with the number of subject discriminators b, which is originally determined to 5 according to the research protocols regarding different acquisition conditions in MASS dataset. We change the value to multiples of 5 so that the subjects belonging to the same subsets are still in the same groups. Table V(b) presents the staging performance with different number of subject discriminators. When we increase b to 40, the overall performance degrades rapidly. We found the number of training parameters also increased significantly, and it was difficult to avoid over-fitting problems. We observe that the number of subject discriminators needs to be carefully determined, considering the model parameters and the training data.

4) Performance Analysis of Hyper Parameter Adjustments: Table V(c) illustrates the performances with different combinations of p in (17) and K as an update cycle. The combinations were chosen within $p = \{2, 3, 4\}$ and $K = \{1, 3, 5, 7\}$ to verify the impacts on training. The best performance was 80.7% for Acc. when p and K were 3. This result implies that the unstable ratio may affect the values of the hyperparameters frequently and degrades performance.

5) Performance Analysis of Sample Balance Between Domains: We reveal that the numbers of source and target samples involved in the training can significantly affect the performance. On the one hand, a network would choose more source samples for training because these samples are usually greater than those of the target samples, and a network is exposed more to source samples during random selection. On the other hand, a network can adopt more target samples because it likely improves the performance in the corresponding domain. However, in our experimental results, none of the strategies are the right choice. In Table V(d), r is defined as the ratio between the numbers of source to target samples chosen during training. The proposed method provides the best performance of around 80.7% when

TABLE V

DIFFERENT ABLATION STUDIES ON (A): EFFECTS OF THE LOCAL DISCRIMINATORS, (B): EFFECTS OF THE NUMBER OF SUBJECT DISCRIMINATORS, (C): ADAPTIVE HYPERPARAMETER ADJUSTMENTS, (D): EFFECTS OF THE BALANCE OF DATA SAMPLES FROM DIFFERENT DOMAINS IN PROP (LEFT) AND FT (RIGHT), RESPECTIVELY. r AND s ARE THE RATIOS

(a)

D^{dom}	D^{stg}	D^{sub}	Acc.
✓	-	-	75.3
✓	✓	-	75.4
✓	-	✓	77.8
✓	✓	✓	80.7

(b)

b	param	Acc.
5	29.0M	80.7
10	31.6M	75.6
20	36.8M	58.8
40	47.4M	42.4

(c)

K	p=2	p=3	p=4
1	79.8	78.5	76.9
3	79.2	80.7	78.4
5	77.1	77.7	76.6
7	77.9	77.2	76.9

(d)

r	Acc.	s	Acc.
0.5	76.7	1	76.3
1	80.7	2	76.3
2	77.0	4	79.1
4	74.8	8	79.3

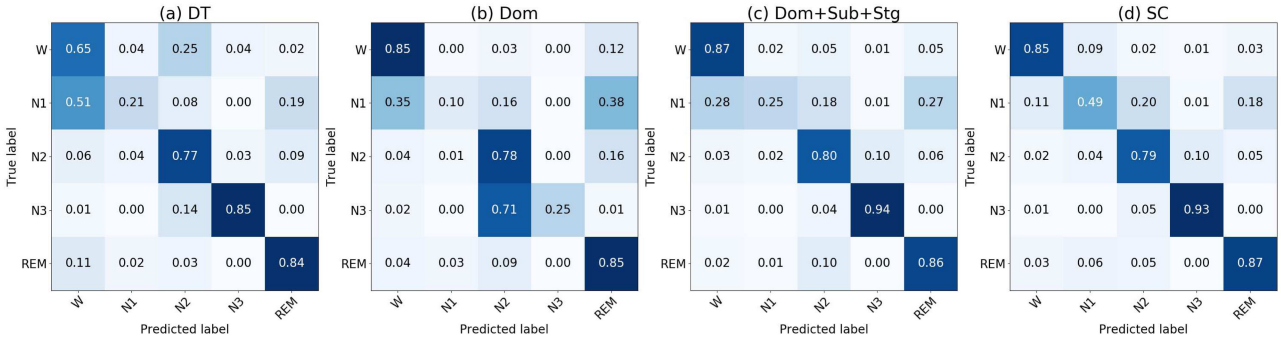

Fig. 6. Confusion matrix results when different combination of the discriminator is used. (a): DT, (b): PROP using only domain discriminator, (c): PROP with domain, subject and stage discriminators, (d): SC. Larger diagonal components of the matrix mean good staging performance. From (a) to (c), the matrix gets resembling (d) which is the upper bound.

the numbers of samples are the same. However, the performance drops by around 3–6% when the ratio is not equal to 1. These results demonstrate that if the number of samples in either domain is excessive, the network can be too fitted to one domain and limited for the other domain. For instance, even though the target samples are more available, the limited knowledge transfer from a small number of source samples can degrade performance. In Table V(d), we also change the ratio $s = \frac{N_s}{N_t}$ in FT [27] and present the results when N_t is fixed to 19. We observe the accuracies are degraded when N_s is smaller. It indicates that the performance of the FT highly depends on the number of source samples.

6) Time Complexity: For training time, the proposed method takes 0.79 ms to train a single sleep epoch of the Sleep-EDF on average, when using DeepSleepNet. In comparison, the original DeepSleepNet takes 0.72 ms. The reason is the additional discriminators. However, the inference time of a single sleep epoch is almost the same of 0.27 ms because the discriminators are not involved during testing.

VI. CONCLUSION

We proposed an unsupervised domain adaptation to effectively utilize DL-based sleep staging networks on small cohorts of sleep data. We used several discriminators to consider not only the global alignment but also local alignments to maintain the intrinsic structures of sleep data. The experimental results demonstrated that the proposed method could achieve similar performance to supervised learning cases even though the network does not need any labels in the target domain. We expect the usefulness of the proposed method in wider scenarios where data acquisition is more difficult than the standard PSG, such as data from wearable devices. Demographic features will be also useful for subject grouping. Further studies are thus needed to prove the validity of proposed method in such situations. Furthermore, the usefulness of this method can extend to other clinical applications using biosignal processing, considering that clinical data acquisition is relatively tough. More studies are needed to replicate this method in other clinical applications to validate its utility.

REFERENCES

[1] R. E. Dahl and D. S. Lewin, "Pathways to adolescent health sleep regulation and behavior," *J. Adolesc. Health*, vol. 31, no. 6, pp. 175–184, 2002.
[2] S. Khalighi, T. Sousa, G. Pires, and U. Nunes, "Automatic sleep staging: A computer assisted approach for optimal combination of features and polysomnographic channels," *Expert Syst. Appl.*, vol. 40, no. 17, pp. 7046–7059, 2013.
[3] S. J. Redmond and C. Heneghan, "Cardiorespiratory-based sleep staging in subjects with obstructive sleep apnea," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 3, pp. 485–496, Mar. 2006.
[4] A. Rechtschaffen, "A manual of standardized terminology, technique and scoring system for sleep stages of human subjects," *Arch. Gen. Psychiatry*, vol. 20, no. 2, pp. 246–247,1969.
[5] C. Iber, I. S. Ancoli, and S. F. Quan, "The AASM manual for the scoring of sleep and associated events: rules, Terminology Technical Specification," *American Academy Sleep Medicine*, 2007.
[6] H. J. Kim *et al.*, "A double-blind, randomized, placebo-controlled crossover clinical study of the effects of alpha-s1 casein hydrolysate on sleep disturbance," *Nutrients*, vol. 11, no. 7, p. 1466, 2019.
[7] J. B. Stephansen *et al.*, "Neural network analysis of sleep stages enables efficient diagnosis of narcolepsy," *Nature Commun.*, vol. 9, no. 1, pp. 1–15, 2018.
[8] K. Mikkelsen and M. De Vos, "Personalizing deep learning models for automatic sleep staging," 2018, *arXiv:1801.02645*.

[9] S. Chambon, M. N. Galtier, P. J. Arnal, G. Wainrib, and A. Gramfort, "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, Apr. 2018.

[10] A. Supratak, H. Dong, C. Wu, and Y. Guo, "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, Nov. 2017.

[11] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos, "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, Mar. 2019.

[12] A. N. Olesen, J. A. Christensen, H. B. Sorensen, and P. J. Jennum, "A noise-assisted data analysis method for automatic EOG-based sleep stage classification using ensemble learning," in *Proc. 38th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2016, pp. 3769–3772.

[13] X. Huang, K. Shirahama, F. Li, and M. Grzegorzek, "Sleep stage classification for child patients using deconvolutional neural network," *Artif. Intell. Med.*, vol. 110, 2020, Art. no. 101981.

[14] H. Dong, A. Supratak, W. Pan, C. Wu, P. M. Matthews, and Y. Guo, "Mixed neural network approach for temporal sleep stage classification," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 2, pp. 324–333, Feb. 2018.

[15] A. H. Ansari *et al.*, "A convolutional neural network outperforming state-of-the-art sleep staging algorithms for both preterm and term infants," *J. Neural Eng.*, vol. 17, no. 1, 2020, Art. no. 016028.

[16] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos, "Automatic sleep stage classification using single-channel EEG: Learning sequential features with attention-based recurrent neural networks," in *Proc. 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2018, pp. 1452–1455.

[17] H. Sun *et al.*, "Sleep staging from electrocardiography and respiration with deep learning," *Sleep*, vol. 43, no. 7, p. zsz306, 2020.

[18] H. Korkalainen *et al.*, "Deep learning enables sleep staging from photoplethysmogram for patients with suspected sleep apnea," *Sleep*, vol. 43, 875, no. 11, p. zsaa098, 2020.

[19] C. Sun, C. Chen, W. Li, J. Fan, and W. Chen, "A hierarchical neural network for sleep stage classification based on comprehensive feature learning and multi-flow sequence learning," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 5, pp. 1351–1366, May 2020.

[20] W. Qu *et al.*, "A residual based attention model for EEG based sleep staging," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 10, pp. 2833–2843, Oct. 2020.

[21] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos, "Joint classification and prediction CNN framework for automatic sleep stage classification," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 5, pp. 1285–1296, May 2019.

[22] H. Seo, S. Back, S. Lee, D. Park, T. Kim, and K. Lee, "Intra-and inter-epoch temporal context network (IITNET) using sub-epoch features for automatic sleep scoring on raw single-channel EEG," *Biomed. Signal Process. Control*, vol. 61, 2020, Art. no. 102037.

[23] C. O'reilly, N. Gosselin, J. Carrier, and T. Nielsen, "Montreal archive of sleep studies: An open-access resource for instrument benchmarking and exploratory research," *J. Sleep Res.*, vol. 23, no. 6, pp. 628–635, 2014.

[24] N. Cooray, F. Andreotti, C. Lo, M. Symmonds, M. T. Hu, and M. De Vos, "Detection of REM sleep behaviour disorder by automated polysomnography analysis," *Clin. Neurophysiol.*, vol. 130, no. 4, pp. 505–514, 2019.

[25] 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, 2019, Art. no. e12786.

[26] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1180–1189.

[27] H. Phan *et al.*, "Towards more accurate automatic sleep staging via deep transfer learning," *IEEE Trans. Biomed. Eng.*, vol. 68, no. 6, pp. 1787–1798, Jun. 2021.

[28] M. Abou Jaoude *et al.*, "Expert-level automated sleep staging of long-term scalp electroencephalography recordings using deep learning," *Sleep*, vol. 43, no. 11, p. zsaa112, 2020.

[29] N. Banluesombatkul *et al.*, "MetaSleepLearner: A pilot study on fast adaptation of bio-signals-based sleep stage classifier to new individual subject using meta-learning," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 6, pp. 1949–1963, Jun. 2021.

[30] M. Shao, C. Castillo, Z. Gu, and Y. Fu, "Low-rank transfer subspace learning," in *Proc. IEEE 12th Int. Conf. Data Mining*, 2012, pp. 1104–1109.

[31] Y. Grandvalet *et al.*, "Semi-supervised learning by entropy minimization," *CAP*, 2005, pp. 281–296.

[32] C.-A. Hou, Y.-H. H. Tsai, Y.-R. Yeh, and Y.-C. F. Wang, "Unsupervised domain adaptation with label and structural consistency," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5552–5562, Dec. 2016.

[33] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096–2030, 2016.

[34] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3934–3941.

[35] L. Hu, M. Kan, S. Shan, and X. Chen, "Unsupervised domain adaptation with hierarchical gradient synchronization," in *Proc. IEEE/CVF Conf. Comput. Vis. Patter Recognit.*, 2020, pp. 4043–4052.

[36] Y. Gu, Z. Ge, C. P. Bonnington, and J. Zhou, "Progressive transfer learning and adversarial domain adaptation for cross-domain skin disease classification," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 5, pp. 1379–1393, May 2020.

[37] K. Zhang, J. Wang, C. W. de Silva, and C. Fu, "Unsupervised cross-subject adaptation for predicting human locomotion intent," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 3, pp. 646–657, Mar. 2020.

[38] H. Korkalainen *et al.*, "Accurate deep learning-based sleep staging in a clinical population with suspected obstructive sleep apnea," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 7, pp. 2073–2081, Jul. 2020.

[39] S. Nasiri and G. D. Clifford, "Attentive adversarial network for large-scale sleep staging," in *Proc. Mach. Learn. Healthcare Conf.*, 2020, pp. 457–478.

[40] C. Sahlin, K. A. Franklin, H. Stenlund, and E. Lindberg, "Sleep in women: Normal values for sleep stages and position and the effect of age, obesity, sleep apnea, smoking, alcohol and hypertension," *Sleep Med.*, vol. 10, no. 9, pp. 1025–1030, 2009.

[41] S. Redline, H. L. Kirchner, S. F. Quan, D. J. Gottlieb, V. Kapur, and A. Newman, "The effects of age, sex, ethnicity, and sleep-disordered breathing on sleep architecture," *Arch. Intern. Med.*, vol. 164, no. 4, pp. 406–418, 2004.

[42] B. Kemp, A. H. Zwinderman, B. Tuk, H. A. Kamphuisen, and J. J. Oberye, "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, Sep. 2000.

[43] A. L. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[44] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, 1960.

[45] A. Khan, J. Sung, and J. W. Kang, "Multi-channel fusion convolutional neural network to classify syntactic anomaly from language-related ERP components," *Inf. Fusion*, vol. 52, pp. 53–61, 2019.