Self-Supervised Learning for Label-Efficient Sleep Stage Classification: A Comprehensive Evaluation

Emadeldeen Eldele, Mohamed Ragab, *Member, IEEE*, Zhenghua Chen, *Senior Member, IEEE*, Min Wu, *Senior Member, IEEE*, Chee-Keong Kwoh, and Xiaoli Li, *Senior Member, IEEE*

***Abstract*— The past few years have witnessed a remarkable advance in deep learning for EEG-based sleep stage classification (SSC). However, the success of these models is attributed to possessing a massive amount of *labeled* data for training, limiting their applicability in real-world scenarios. In such scenarios, sleep labs can generate a massive amount of data, but labeling can be expensive and time-consuming. Recently, the self-supervised learning (SSL) paradigm has emerged as one of the most successful techniques to overcome labels' scarcity. In this paper, we evaluate the efficacy of SSL to boost the performance of existing SSC models in the few-labels regime. We conduct a thorough study on three SSC datasets, and we find that fine-tuning the pretrained SSC models with only 5% of labeled data can achieve competitive performance to the supervised training with full labels. Moreover, self-supervised pretraining helps SSC models to be more robust to data imbalance and domain shift problems.**

***Index Terms*— Sleep stage classification, EEG, self-supervised learning, label-efficient learning.**

Manuscript received 16 October 2022; revised 2 January 2023 and 30 January 2023; accepted 11 February 2023. Date of publication 14 February 2023; date of current version 21 February 2023. This work was supported in part by the Ministry of Education (MOE) Academic Research Fund Tier 2 under the project "Host-pathogen protein-protein interaction approaches for predicting virulence" under Grant MOE2019-T2-2-175, in part by the Agency for Science, Technology, and Research (A*STAR) under its Advanced Manufacturing and Engineering (AME) Programmatic Funds under Grant A20H6b0151, and in part by the Career Development Award under Grant C210112046. The work of Emadeldeen Eldele was supported by the A*STAR Singapore International Graduate Award (SINGA) Scholarship. *(Corresponding author: Min Wu.)*

Emadeldeen Eldele and Chee-Keong Kwoh are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: emad0002@ntu.edu.sg; asckkwoh@ntu.edu.sg).

Mohamed Ragab and Zhenghua Chen are with the Centre for Frontier AI Research (CFAR), Institute for Infocomm Research (I^2R), Agency for Science, Technology and Research (A*STAR), Singapore 138632 (e-mail: mohamedr002@e.ntu.edu.sg; chen0832@e.ntu.edu.sg).

Min Wu is with the Institute for Infocomm Research (I^2R), Agency for Science, Technology and Research (A*STAR), Singapore 138632 (e-mail: wumin@i2r.a-star.edu.sg).

Xiaoli Li is with the Centre for Frontier Research (CFAR), Institute for Infocomm Research (I^2R), Agency of Science, Technology and Research (A*STAR), Singapore 138632, and also with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: xlli@i2r.a-star.edu.sg).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNSRE.2023.3245285, provided by the authors.

Digital Object Identifier 10.1109/TNSRE.2023.3245285

I. Introduction

Sleep stage classification (SSC) plays a key role in diagnosing many common diseases such as insomnia and sleep apnea [1]. To assess the sleep quality or diagnose sleep disorders, overnight polysomnogram (PSG) readings are split into 30-second segments, i.e., epochs, and assigned a sleep stage. This process is performed manually by specialists, who follow a set of rules, e.g., the American Academy of Sleep Medicine (AASM) [2] to identify the patterns and classify the PSG epochs into sleep stages. This manual process is tedious, exhaustive, and time-consuming.

To overcome this issue, numerous deep learning-based SSC models were developed to automate the data labeling process. These models are trained on a massive labeled dataset and applied to the dataset of interest. For example, Jadhav et al. [3] explored different deep learning models to exploit raw electroencephalogram (EEG) signals, as well as their time-frequency spectra. Also, Phyo et al. [4] attempted to improve the performance of the deep learning model on the confusing transitioning epochs between stages. In addition, Phan et al. [5] proposed a transformer backbone that provides interpretable and uncertainty-quantified predictions. However, the success of these approaches hinges on a massive amount of *labeled* data to train the deep learning models, which might not be feasible. In practice, sleep labs can collect a vast amount of overnight recordings, but the difficulties in labeling the data limit deploying these data-hungry models. Thus, unfortunately, the SSC works developed in the past few years have now a bottleneck: the size, quality, and availability of labeled data.

One alternative solution to pass through this bottleneck is the self-supervised learning (SSL) paradigm, which witnessed increased interest recently due to its ability to learn useful representations from unlabeled data. In SSL, the model is pretrained on a newly defined task that does not require any labeled data, where ground-truth pseudo labels can be generated for free. Such tasks are designed to learn the model to recognize general characteristics about the data without being directed with labels. Currently, SSL algorithms can produce state-of-the-art performance on standard computer vision benchmarks [6], [7], [8], [9]. Consequently, the SSL paradigm has gained more interest to be applied for sleep stage classification problem [10], [11].

Most prior works aim to propose novel SSL algorithms and show how they could improve the performance of sleep stage classification. Instead, in this work, our aim is to examine the efficacy of the SSL paradigm to re-motivate deploying existing SSC works in real-world scenarios, where only few-labeled samples are available. Therefore, we revisit a prominent subset of SSC models and perform an empirical study to evaluate their performance under the few-labeled data settings. Moreover, we explore the efficacy of different SSL algorithms on their performance and robustness. We also study the effect of sleep data characteristics, e.g., data imbalance and temporal relations, on the learned self-supervised representations. Finally, we assess the transferability of self-supervised against supervised representations and their robustness to domain shift. The overall framework is illustrated in Fig. 1. We perform an extensive set of experiments on three sleep staging datasets to systemically analyze the SSC models under the few-labeled data settings. The experimental results of this study aim to provide a solid and realistic real-world assessment of the existing sleep stage classification models.

To summarize, our contributions are as follows:

1) We provide a systematic evaluation of the self-supervised learning for sleep stage classification models under the few-label settings. We aim to inspire deploying existing SSC models in real-world data-scarce scenarios.
2) We provide an empirical study to assess the robustness of self-supervised learning against sleep data imbalance and domain-shift problems.
3) We provide recommendations for deploying and improving the self-supervised learning algorithms for sleep stage classification.
4) Our code is publicly available for more practical implementation of different SSC models with other SSL algorithms.

II. Related Work

A. Sleep Stage Classification

A wide span of EEG-based sleep stage classification methods have been introduced in recent years. These methods proposed different architectural designs. For example, some methods adopted multiple parallel convolutional neural networks (CNNs) branches to extract better features from EEG signals [4], [12], [13]. Also, some methods included residual CNN layers [15], [16], while others used graph-based CNN networks [17]. On the other hand, Phan et al. [18] proposed Long Short Term Memory (LSTM) networks to extract features from EEG spectrograms. To handle the temporal dependencies among EEG features, these methods had different approaches. For instance, some works adopted recurrent neural networks (RNNs), e.g., bi-directional LSTM networks as in [12], [16], and [4]. Other works adopted the multi-head self-attention as a faster and more efficient way to capture the temporal dependencies in timesteps, as in [19] and [13].

Despite the proven performance of these architectures, they require a huge labeled training dataset to feed the deep learning models. None of these works studied the performance of their models in the few-labeled data regime, which is our scope in this work.

B. Self-Supervised Learning Approaches

Self-supervised learning received more attention recently because of its ability to learn useful representations from unlabeled data. The first SSL auxiliary tasks showed a big improvement in the performance of the downstream task. For example, Noroozi et al. proposed training the model to solve a jigsaw puzzle on a patched image [20]. In addition, Gidaris et al. proposed rotating the input images, then trained the model to predict the rotation angle [21]. The success of these auxiliary tasks motivated adapting contrastive learning algorithms, which showed to be more effective due to their ability to learn invariant features. The key idea behind contrastive learning is to define positive and negative pairs for each sample, then push the sample closer to the positive pairs, and pull it away from the negative pairs. In general, contrastive-based approaches rely on data augmentations to generate positive and negative pairs. For example, SimCLR considered the augmented views of the sample as positive pairs, while all the other samples within the same mini-batch are considered as negative pairs [6]. Also, MoCo increased the number of negative pairs by keeping samples from other mini-batches in a memory bank [7]. On the other hand, some recent algorithms neglected the negative pairs and proposed using only positive pairs such as SimSiam [8] and BYOL [9].

C. Self-Supervised Learning for Sleep Staging

The success of SSL in computer vision applications motivated their adoption for sleep stage classification. For example, Mohsenvand et al. [22] and Jiang et al. [23] proposed SimCLR-like methodologies and applied EEG-related augmentations for sleep stage classification. Also, Banville et al. applied three pretext tasks, i.e., relative positioning, temporal shuffling, and contrastive predictive coding (CPC) to explore the underlying structure of the unlabeled sleep EEG data [24]. The CPC [14] algorithm predicts the future timesteps in the time-series signal, which motivated other works to build on it. For example, SleepDPC solved two problems, i.e., predicting future representations of epochs, and distinguishing epochs from other different epochs [25]. Also, TS-TCC proposed temporal and contextual contrasting approaches to learn instance-wise representations about the sleep EEG data [10]. In addition, SSLAPP developed a contrastive learning approach with attention-based augmentations in the embedding space to add more positive pairs [26]. Last, CoSleep [11] and SleepECL [27] are yet another two contrastive methods that exploit information, e.g., inter-epoch dependency and frequency domain views, from EEG data to obtain more positive pairs for contrastive learning.

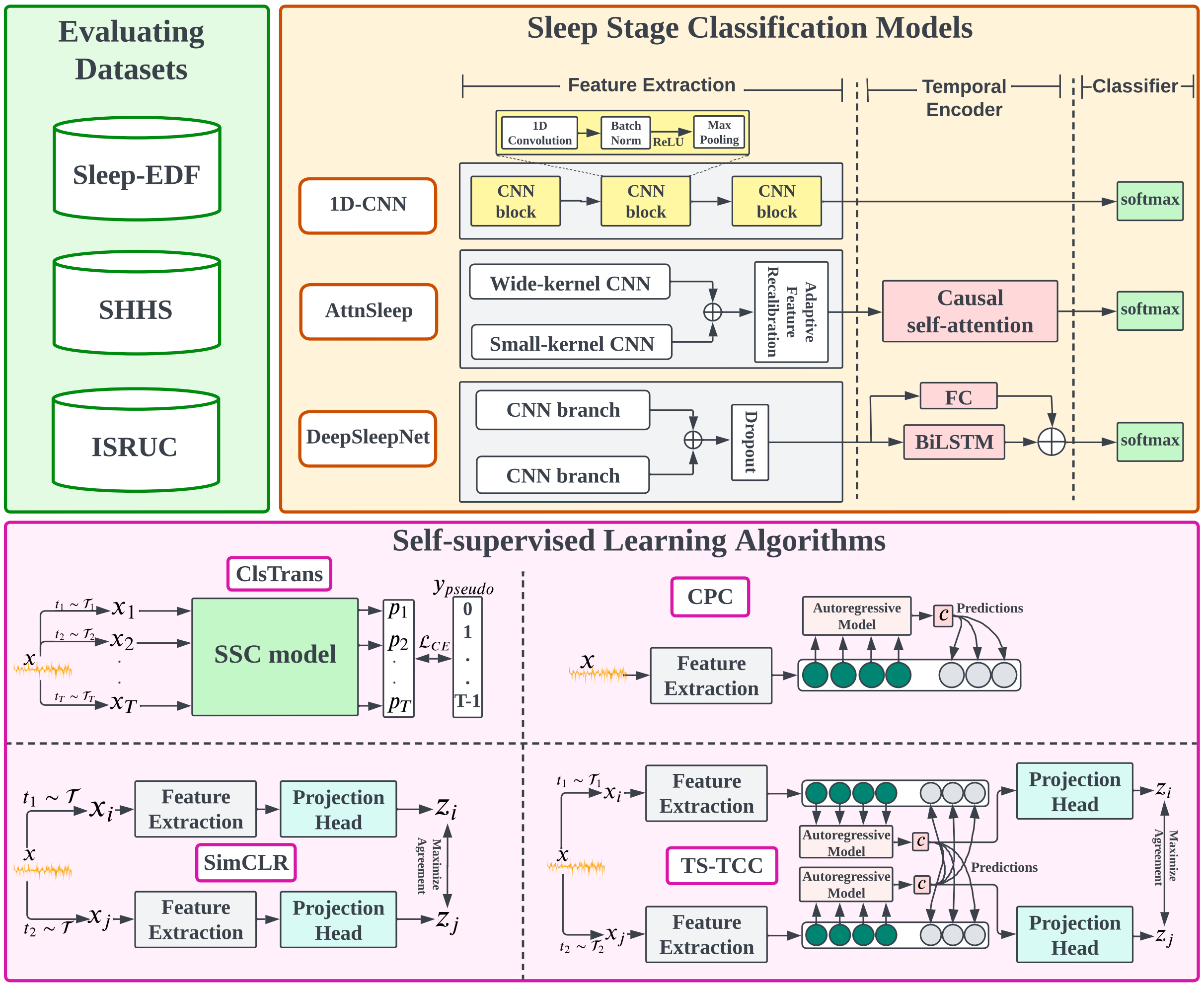

Fig. 1. The architecture of our evaluation framework. We experiment with three sleep stage classification models, i.e., DeepSleepNet [12], AttnSleep [13], and 1D-CNN [10]. We also include four self-supervised learning algorithms, i.e., ClsTran, SimCLR [6], CPC [14], and TS-TCC [10]. The different experiments are performed on Sleep-EDF, SHHS, and ISRUC datasets.

III. Evaluation Framework

A. Preliminaries

In this section, we describe the SSL-related terminologies, i.e., pretext tasks, contrastive learning, and downstream tasks.

1) Problem Formulation: We assume that the input is single-channel EEG data in $\mathbb{R}^d$, and each sample has one label from one of C classes. The supervised downstream task has an access to the inputs and the corresponding labels, while the self-supervised learning algorithms have access only to the inputs.

The SSC networks consist of three main parts. The first is the feature extractor, which maps the input data into the embedded space $f_\phi : \mathbb{R}^d \rightarrow \mathbb{R}^{m1}$ parameterized by neural network parameters ϕ. The second is the temporal encoder (TE), which is another intermediate network to improve the temporal representations. The TE may change the dimension of the embedded features $f_\theta : \mathbb{R}^{m1} \rightarrow \mathbb{R}^m$. Finally, the classifier $f_\gamma : \mathbb{R}^m \rightarrow \mathbb{R}^C$, which produces the predictions. The SSL algorithms learn ϕ from unlabeled data, while fine-tuning learns θ and γ with also updating ϕ.

2) Pretext Tasks: Pretext tasks refer to the pre-designed tasks to learn the model generalized representations from the unlabeled data. Here, we describe two main types of pretext tasks, i.e., auxiliary, and contrastive tasks.

a) Auxiliary tasks: This category includes defining a new task along with free-to-generate pseudo labels. These tasks can be defined as classification, regression, or any others. In the context of time-series applications, a new classification auxiliary task was defined in [28] and [29] by generating several views to the signals using augmentations, e.g., adding noise, rotation, and scaling. These augmentations are meant to improve the robustness of the model against the different variations that could be applied to the signals. For example, adding random noise to the original signal helps the model become robust against the noisy samples. Similarly, scaling the signal magnitude by a random scalar allows the model to become robust against amplitude and offset invariances. Last, rotating the signal by inverting its sign simulates some situations where the sensor is held upside down, which helps the model to robust against sensor-placement invariance. To train this auxiliary task, each view is assigned a label, and the model was pretrained to classify these transformations. This approach showed success in learning underlying representations from unlabeled data. However, it is usually designed with heuristics that might limit the generality of the learned representations [10].

b) Contrastive learning: In contrastive learning, representations are learned by comparing the similarity between samples. In specific, we define positive and negative pairs for each sample. Next, the feature extractor is trained to achieve the contrastive objective, i.e., push the features of the sample towards the positive pairs, and pull them away from the negative pairs. These pairs are usually generated via data augmentations. Notably, some studies [6], [30] relied on strong successive augmentations and found them to be a key factor in the success of their contrastive techniques.

Formally, given a dataset with N unlabeled samples, we generate two views for each sample $\mathbf{x}$, i.e., $\{\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j\}$ using data augmentations. Therefore, in a multi-viewed batch with N samples for each view, we have a total of $2N$ samples. Next, the feature extractor transforms them into the embedding space, and a projection head $h(\cdot)$ is used to obtain low-dimensional embeddings, i.e., $\mathbf{z}_i = h(f_\phi(\hat{\mathbf{x}}_i))$ and $\mathbf{z}_j = h(f_\phi(\hat{\mathbf{x}}_j))$. Assuming that for an anchor sample indexed $i \in I \equiv \{1 \ldots 2N\}$, and $A(k) \equiv I \backslash \{k\}$. The objective of contrastive learning is to encourage the similarity between positive pairs and separate the negative pairs apart using the NT-Xent loss, defined as follows:

$$\mathcal{L}_{\text{NT-Xent}} = \frac{-1}{2N} \sum_{i \in I} \log \frac{\exp\left(\mathbf{z}_i \cdot \mathbf{z}_j / \tau\right)}{\sum_{a \in A(i)} \exp\left(\mathbf{z}_i \cdot \mathbf{z}_a / \tau\right)}, \quad (1)$$

where $\cdot$ symbol denotes the inner dot product, and τ is a temperature parameter.

3) Downstream Tasks: Downstream tasks are the main tasks of interest that lacked a sufficient amount of labeled data for training the deep learning models. In this paper, the downstream task is sleep stage classification, i.e., classifying the PSG epochs into one of five classes, i.e., W, N1, N2, N3, and REM. However, in general, the downstream task can be different and defined by various applications. Notably, different pretext tasks can have a different impact on the same downstream task. Therefore, it is important to design a relevant pretext task to the problem of interest, to learn better representations. Despite the numerous proposed methods in self-supervised learning, identifying the proper pretext task is still an open research question [31].

B. Sleep Stage Classification Models

We perform our experiments on three sleep stage classification models, i.e., DeepSleepNet [12], AttnSleep [13], and 1D-CNN [10]. The architectures of these models are shown in Fig 1. Each model has its specifically-designed feature extractor, temporal encoder, and methodology to address the sleep data imbalance issue. Next, we discuss each SSC model in more details.

1) DeepSleepNet: DeepSleepNet consists of two parallel convolutional network branches with dropout to extract features. These features are passed to the temporal encoder that contains a Bidirectional Long Shot Term Memory (BiLSTM) network with a residual connection. To overcome the data imbalance issue in sleep data and achieve good performance in minor classes, DeepSleepNet is trained in two separate phases. In the first, the model is trained with oversampled balanced data, while in the second, the pretrained model is fine-tuned with the original imbalanced data.

2) AttnSleep: AttnSleep extracts features from EEG data with a multi-resolution CNN network followed by an adaptive feature recalibration module. The extracted features are then sent to a causal self-attention network to characterize the temporal relations. AttnSleep deploys a class-aware loss function to handle the class imbalance issue. This loss function assigns different weights to the data based on two factors, i.e., the distinctness of the features of each class, and the number of samples of that class in the dataset.

3) 1D-CNN: The 1D-CNN network consists of three convolutional blocks. Each block consists of a 1D-Convolutional layer followed by a BatchNorm layer, a non-linearity ReLU activation function, and a MaxPooling layer. This architecture does not include any special component to find the temporal relations nor handle the data imbalance issue in sleep EEG data.

In our experiments, we pretrain only the feature extractor of the three SSC models. After that, we fine-tune the whole model with the few-labeled data in an end-to-end manner.

C. Self-Supervised Learning Algorithms

In this section, we describe the adopted SSL algorithms (see Fig. 1) in more details. We selected four algorithms that can be applied to any feature extractor design.

1) ClsTran: Classifying Transformations is an auxiliary classification task, in which we first apply some transformations to the input signal. Then, we associate an automatically-generated pseudo label with each transformation. Last, we train the model to classify the transformed signals based on these pseudo labels.

Formally, let's assume a tuple of an input signal and its corresponding pseudo label $(x_i, \hat{y}_i)$, where x_i is i^{th} transformed signal, $\hat{y}_i$ is the generated pseudo label that corresponds to the k^{th} transformation, and $k \in [0, T)$, T is the total number of transformations. Next, the transformed signal passes through the feature extractor, the temporal encoder, and the classifier networks to generate the output probability p_t. Last, the model is trained to minimize a standard cross-entropy loss based on these pseudo labels: $\mathcal{L}_{\text{CE}} = \sum_{t=0}^{T-1} \mathbb{1}_{[\hat{y}=t]} \log p_t$, where $\mathbb{1}$ is the indicator function, which is set to be 1 when the condition is met, and set to 0 otherwise. In this work, we adopt four augmentations, i.e., negation, permutation, adding noise, and time shifting, which were adopted by previous works [10], [22] and showed good downstream performance. More details about data augmentations are provided in Section SII in the supplementary materials.

2) SimCLR: Simple framework for Contrastive Learning of Visual Representation [6] is a contrastive SSL algorithm that relies on data augmentations to learn invariant representations. It consists of four major components. The first is data augmentations, which are utilized to generate two correlated views of the same sample. The second is the feature extractor network that transforms the augmented views into latent space. The third is the projection head, which maps the features into a low-dimensional space. The fourth is the NT-Xent loss (Eq. 1), which aims to maximize the similarity between an anchor sample with its augmented views while minimizing its similarity with the augmented views of the other samples within the mini-batch.

3) CPC: Contrastive Predictive Coding [14] is a predictive contrastive SSL approach that learns representations of time-series signals by predicting the future timesteps in the embedding space. To do so, the feature extractor first generates the latent feature embeddings for the input signals. Next, an autoregressive model receives a part of the embeddings, i.e., the past timesteps, then generates a context vector and uses it to predict the other part, i.e., the future timesteps. CPC deploys a contrastive loss such that the embedding should be close to positive future embeddings and distant from negative future embeddings. CPC showed improved downstream performance in various time-series and speech recognition-related tasks, without the need for any data augmentation.

4) TS-TCC: Time-Series representation learning via Temporal and Contextual Contrasting [10] is yet another contrastive SSL approach for time-series data. TS-TCC relies on strong and weak augmentations to generate two views of an anchor sample. Next, the feature embeddings of these views are generated. Next, similar to CPC, a part of the embeddings of each view is sent to an autoregressive model to generate a context vector. Then, the context vector generated for one augmented view is used to predict the future timesteps of the other augmented view with a contrastive loss. Therefore, it pushes the embeddings of one augmented view to the positive future embeddings of the other augmented view, and vice versa. In addition, it leverages the NT-Xent loss (Eq. 1) to maximize the agreement between the context vectors of the same sample, while maximizing it within the contexts of other samples.

IV. Experimental Setup

A. Datasets

We evaluate the SSL algorithms on three sleep stage classification datasets, namely Sleep-EDF, SHHS, and ISRUC. These datasets have different characteristics in terms of sampling rates, EEG channels, and the health conditions of subjects. We use a single EEG channel from each dataset in our experiments following previous works [12], [13].

1) Sleep-EDF: Sleep-EDF dataset [32] is a public dataset that contains the polysomnography (PSG) readings of 20 healthy subjects (10 males and 10 females). In our experiments, we adopted the recordings included in the Sleep Cassette (SC) study and used the EEG data from Fpz-Cz channel with a sampling rate of 100 Hz.

2) SHHS: Sleep Heart Health Study [33], [34] is a multi-center cohort study of the cardiovascular and other consequences of sleep-disordered breathing. The dataset is created to record the PSG readings of patients aged 40 years and older in two visits. In our experiments, we randomly chose 20 subjects from the patients during the first visit and chose the EEG channel C4-A1 with a sampling rate of 125 Hz.

3) ISRUC: ISRUC dataset [35] contains PSG recordings for human adults with different health conditions. We selected the 10 healthy subjects included in subgroup III and extracted the EEG channel C4-A1 with a sampling rate of 200 Hz.

More details about the datasets are provided in Table I.

TABLE I
Details of the Three Datasets Used in Our Experiments (Each Sample Is a 30-second Epoch). S.R. Refers to the Sampling Rate

Datasets	Channel	S.R.	W	N1	N2	N3	REM	#Total
Sleep-EDF	Fpz-Cz	100 Hz	8285 *19.6%*	2804 *6.6%*	17799 *42.1%*	5703 *13.5%*	7717 *18.2%*	42308
SHHS	C4-A1	125 Hz	4741 *23.7%*	783 *3.9%*	8204 *40.9%*	2747 *13.7%*	3546 *17.8%*	20021
ISRUC	C4-A1	200 Hz	1817 *20.4%*	1248 *14.0%*	2678 *30.1%*	2035 *22.9%*	1111 *12.5%*	8889

B. Implementation Details

1) Dataset Preprocessing: For all the datasets, we apply the two preprocessing steps. First, we only considered the five sleep stages according to the AASM standard. Second, we exclude the wake periods that exceed 30 minutes before and after the sleep periods following [12] and [13]. We also split the subjects into five folds, and *all* the upcoming experiments are performed with 5-fold subject-wise cross-validation.

2) Training Scheme: Following TS-TCC [10], the pretraining, as well as the fine-tuning, were performed for 40 epochs with a batch size of 128. The neural network weights were optimized using the Adam optimizer, with a learning rate of 1e-3 and a weight decay of 1e-4. We reported the results in terms of accuracy and macro F1-score. We fixed the training parameters to ensure a fair evaluation scheme among all the methods. For TS-TCC, since it is the only SSL algorithm that contains two losses, we used the same weights for the losses as mentioned in the original work (i.e., $\lambda_1 = 1$ and $\lambda_2 = 0.7$). Our codes are built using PyTorch 1.7 and they are publicly available at github.com/emadeldeen24/eval_ssl_ssc.

3) Fine-Tuning: The procedure of self-supervised training on SSC models is further depicted in Fig. 2. It starts with leveraging the feature extractor from the sleep stage classification model, regardless of its architecture, for self-supervised pretraining. Subsequently, the pretrained feature extractor is fine-tuned along with the other SSC model components, i.e., the temporal encoder and classifier, using the few labeled examples. This fine-tuning process allows more effective use of the available labels to boost performance.

V. Results

A. Which SSL Algorithm Performs Best?

In Tables II, we compare the *supervised* performance of the three SSC models (Section III-B) against the *fine-tuned*

TABLE II
RESULTS OF FINE-TUNING PRETRAINED MODELS WITH DIFFERENT SSL TECHNIQUES WITH ONLY 1% OF LABELS (PER-CLASS PERFORMANCE IS IN TERMS OF F1-SCORE). **BEST** RESULTS ARE IN BOLD, WHILE SECOND BEST ARE UNDERLINED

		DeepSleepNet							AttnSleep							1D-CNN						
	Algorithm	W	N1	N2	N3	REM	ACC	MF1	W	N1	N2	N3	REM	ACC	MF1	W	N1	N2	N3	REM	ACC	MF1
Sleep-EDF	Supervised	67.4	24.1	78.0	80.6	51.7	68.2	60.4	71.5	17.1	72.9	79.9	46.5	65.3	57.6	68.0	14.7	73.0	70.8	53.3	64.3	55.9
	ClsTran	60.1	17.2	69.6	73.1	48.1	61.2	53.6	57.4	11.5	65.5	64.8	42.2	57.3	48.3	72.7	12.7	76.6	78.2	47.1	67.9	57.5
	SimCLR	76.7	**25.2**	83.0	79.5	**65.7**	74.8	66.0	67.9	18.6	**80.0**	80.6	**58.7**	70.5	61.2	80.6	19.7	**83.7**	84.4	66.1	76.4	66.9
	CPC	**78.7**	21.3	**84.1**	82.3	61.3	74.8	65.5	72.7	20.0	78.3	78.4	51.6	68.6	60.2	80.3	**20.0**	81.3	80.1	59.0	73.3	64.1
	TS-TCC	78.5	24.5	82.9	**83.2**	63.8	**75.2**	**66.6**	**77.8**	**22.1**	78.8	**83.5**	53.9	**71.2**	**63.2**	**82.7**	19.2	83.2	**84.7**	**66.4**	**76.5**	**67.2**
	Algorithm	W	N1	N2	N3	REM	ACC	MF1	W	N1	N2	N3	REM	ACC	MF1	W	N1	N2	N3	REM	ACC	MF1
SHHS	Supervised	63.7	1.0	73.7	76.2	51.2	66.8	53.2	55.0	3.8	68.8	69.4	46.3	60.3	48.7	43.6	0.5	65.8	59.2	45.7	56.9	42.9
	ClsTran	39.5	0.2	61.5	63.0	34.7	52.9	39.8	59.8	6.7	65.0	59.5	45.8	58.5	47.3	62.9	0.1	69.8	69.7	39.9	62.6	48.5
	SimCLR	**77.3**	**7.8**	77.3	77.8	54.0	**71.9**	**58.8**	71.3	**7.2**	74.6	76.6	**48.6**	68.4	55.6	**78.9**	3.3	**79.4**	**81.7**	**54.6**	**74.2**	**59.6**
	CPC	72.8	4.2	76.0	76.6	**54.2**	69.5	56.8	62.4	5.5	55.9	63.4	36.6	53.6	44.8	76.9	**9.8**	71.6	70.9	50.4	67.6	55.9
	TS-TCC	74.4	3.7	**78.2**	**78.4**	52.8	71.2	57.5	**73.3**	4.6	**74.8**	**77.5**	48.5	**68.9**	**55.7**	77.8	4.8	78.7	80.5	51.2	72.9	58.6
	Algorithm	W	N1	N2	N3	REM	ACC	MF1	W	N1	N2	N3	REM	ACC	MF1	W	N1	N2	N3	REM	ACC	MF1
ISRUC	Supervised	63.6	35.0	44.6	75.5	**46.7**	55.5	53.1	63.7	**38.5**	43.3	73.8	21.3	52.9	48.1	51.7	28.7	27.3	63.9	36.1	46.3	41.5
	ClsTran	49.5	33.6	36.5	64.4	32.5	46.9	43.3	58.1	33.5	26.3	60.3	22.1	41.4	40.0	36.6	27.4	18.4	74.6	31.9	42.7	37.8
	SimCLR	70.0	40.2	49.6	77.9	42.6	58.6	56.1	69.2	37.6	53.8	75.9	30.0	58.7	53.3	76.7	42.1	55.4	76.0	21.0	59.7	54.3
	CPC	76.5	33.7	**62.6**	**80.3**	40.5	**65.1**	58.7	**77.0**	32.2	**57.3**	**78.3**	31.2	**60.6**	**55.2**	77.0	39.6	**58.5**	**85.4**	37.5	63.5	59.6
	TS-TCC	**79.5**	**44.5**	58.1	77.4	42.0	63.6	**60.3**	69.0	36.3	49.0	73.5	**31.8**	54.9	51.9	**82.0**	**42.5**	56.8	84.9	**46.4**	**65.9**	**62.5**

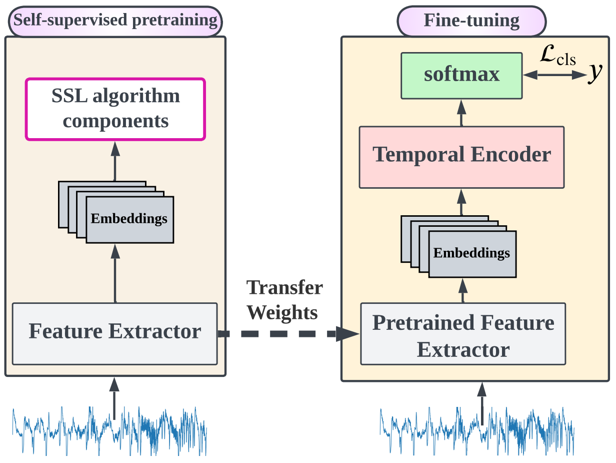

Fig. 2. In self-supervised pretraining, we use the feature extractor of the sleep stage classification model to train the self-supervised task with the unlabeled data. Next, we fine-tune the pretrained feature extractor along with the other sleep stage classification model components with the few labeled data.

models with the four SSL algorithms (Section III-C) using 1% of labeled data.

We notice that self-supervised pretraining with contrastive methods ensures better performance against supervised training in the few-labeled data regime. Specifically, we find that SimCLR, CPC, and TS-TCC demonstrate remarkable performance on the three datasets. This indicates that learning invariant representations by contrastive learning can achieve good generalization on sleep datasets. Counterpart, pretraining with the auxiliary task learns poorer representations, leading to a downgraded performance except for few cases. This could be regarded to the high complexity of sleep EEG data, which does not help the model identify the difference between several augmented views.

We also conducted several experiments to assess the capability of SSL algorithms in learning temporal information, which are provided in the supplementary materials (see Section SIII-C). We find that pretrained models with CPC and TS-TCC can be robust to the existence and the type of temporal encoder while fine-tuning. The reason is that these methods rely on predicting the future timesteps in the latent space, which allows them to learn about temporal features in the EEG data.

B. Performance Under Different Few-Labels Settings

We study the performance of pretrained models when fine-tuned with different amounts of labeled data, i.e., 1%, 5%, 10%, and 100%. Fig. 3 shows the result of these experiments on the Sleep-EDF dataset (results on SHHS and ISRUC datasets are provided in Section SIII-B in the supplementary materials). We find that for the three SSC models, fine-tuning with 5 or 10% of labels can achieve very close performance to the supervised training with 100% of labels. This demonstrates that self-supervised pretrained models yield richer embeddings than their supervised counterparts, which enhances the downstream task performance with such few labels.

Specifically, fine-tuning CPC-pretrained DeepSleepNet with 5% of labeled data could achieve an F1-score of 72.5%, which is only 2.1% less than supervised training with full labels. Also, fine-tuning TS-TCC-pretrained AttnSleep and 1D-CNN with 5% of labeled data had a difference from the fully supervised training of 5.1% and 1.5% respectively. Similarly, fine-tuning with 10% of labeled data has even lower differences of 1.4, 3.4, and 1.3% on DeepSleepNet, AttnSleep, and 1D-CNN respectively with fully supervised training. These results indicate the applicability of existing SSC

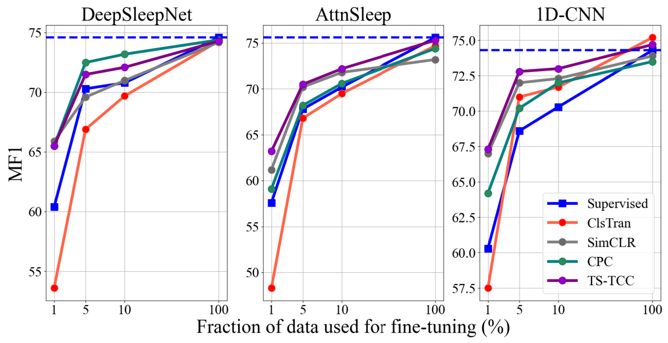

Fig. 3. Fine-tuning the pretrained SSL algorithms with different fractions of labeled Sleep-EDF data.

TABLE III
COMPARISON BETWEEN SOTA SELF-SUPERVISED METHODS FOR SLEEP STAGE CLASSIFICATION AND EXISTING FINE-TUNED SSC MODELS. EXPERIMENTS ARE APPLIED ON SLEEP-EDF DATASET

Method	Data Split	Chanels	Labels%	ACC	MF1
SleepDPC	5-fold CV	1 EEG	10	61.4	55.3
SSLAPP	5-fold CV	1 EEG	10	71.3	58.0
DeepSleepNet + CPC	5-fold CV	1 EEG	10	**81.0**	**73.2**
AttnSleep + TS-TCC	5-fold CV	1 EEG	10	79.8	72.2
1D-CNN + TS-TCC	5-fold CV	1 EEG	10	<u>80.9</u>	<u>73.0</u>

works in real-world scenarios provided the self-supervised pretraining.

We also find that the gain from self-supervised pretraining tends to diminish with fine-tuning the model with the fully labeled data. This observation holds for all three SSC models on the three datasets. Therefore, we can conclude that self-supervised pretraining can provide better regularization, reducing the overfitting problem. However, it does not improve the optimization to reduce the underfitting problem, which is aligned with the findings in [36].

C. Comparison With Baselines

We compare the performance of the adopted pretrained SSC models against state-of-the-art self-supervised methods proposed specifically for the sleep stage classification problem. Table III provides the comparison results, where we show the reported results of SleepDPC [25] and SSLAPP [26] on Sleep-EDF dataset. To have a fair evaluation, we re-implemented these methods to be consistent with our experimental settings. Specifically, unified a 5-fold cross-validation instead of the 20-fold cross-validation and 80%/20% split in SleepDPC and SSLAPP respectively. Also, we included the same single EEG channel as in our experiments instead of using 2 EEG channels or a combination of EEG and EOG channels as in SleepDPC and SSLAPP respectively. We compare these methods against existing SSC models with the best-performing SSL method.

The experimental results show a noticeable advantage of pretraining existing SSC models over state-of-the-art sleep-specific SSL methods in terms of both accuracy and macro F1-score. This could be regarded to the improved performance of existing SSC models that were specifically designed to address different aspects of sleep EEG data. Moreover, plugging a good-performing SSL algorithm with these models can boost their performance under data-scarce scenarios. Therefore, it is important to rebirth existing SSC models with self-supervised learning to obtain comparable results in real-world scenarios.

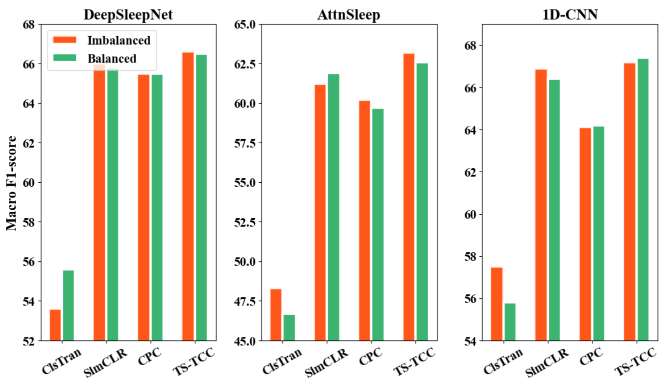

Fig. 4. Comparison between pretraining self-supervised algorithms with original imbalanced dataset vs. oversampled original dataset applied on Sleep-EDF dataset.

D. Robustness of SSL Against Sleep Data Imbalance

The nature of sleep stages implies that some stages, e.g., N1, occur less frequently than other stages such as N2. Consequently, the sleep stage datasets are usually imbalanced (see Table I). Therefore, it is important to study whether the data imbalance affects the quality of the learned representations by SSL algorithms. To do so, we compute the performance gap between models pre-trained on balanced and imbalanced datasets. Specifically, we pretrain the SSL algorithms with the original imbalanced data, and also with oversampled balanced data [12]. The experimental results are shown in Fig. 4.

We observe that the gap between balanced and imbalanced pretraining is minor for *contrastive* SSL algorithms, and it does not exceed a maximum of 0.2%, 0.6%, and 0.5% for DeepSleepNet, AttnSleep, and 1D-CNN respectively. These observations show that contrastive SSL algorithms are more robust to dataset imbalance, which is consistent with previous studies [6], [37]. The main reason is their ability to learn more general and richer features from the majority classes than traditional supervised learning. In specific, the learned self-supervised representations are not supervised or motivated by any labels, i.e., they are not label-directed, and they could learn other intrinsic properties in the EEG signal. These features can improve the classification performance of the minor classes and the learned features can be more efficient for the downstream task. On the other hand, the ClsTran algorithm is directed by a cross-entropy loss that depends on the assigned pseudo labels. Therefore, it can be affected by the data imbalance, and it shows different performance with oversampled data. In the supplementary material, we also analyze the ability of SSL to improve the performance of the minor classes (see Section SIII-A).

TABLE IV

TRANSFERABILITY EXPERIMENT APPLIED ON FIVE CROSS-SUBJECT SCENARIOS. IN THE FIRST TABLE, SOURCE AND TARGET SUBJECTS ARE FROM THE SLEEP-EDF DATASET (ANNOTATED WITH E) AND TRAINING IS PERFORMED WITH 100% OF THE SOURCE DOMAIN LABELS. IN THE SECOND TABLE, WE REDUCED THE SOURCE DOMAIN LABELS TO 1%. IN THE THIRD TABLE, SOURCE DOMAIN SUBJECTS ARE FROM SHHS DATASET (ANNOTATED WITH S), WHILE THE TARGET SUBJECTS ARE FROM SLEEP-EDF DATASET. THE SUPERVISED TRAINING, AS WELL AS THE SSL ALGORITHMS FINE-TUNING IS PERFORMED WITH 100% OF SOURCE DOMAIN LABELS. **BEST** RESULTS ARE IN BOLD, WHILE SECOND BEST ARE UNDERLINED. RESULTS ARE IN TERMS OF MF1-SCORE

	Intra-dataset Evaluation — With 100% of source domain labels																	
	DeepSleepNet						**AttnSleep**						**1D-CNN**					
	E0→E7	E1→E8	E2→E9	E3→E10	E4→E11	AVG	E0→E7	E1→E8	E2→E9	E3→E10	E4→E11	AVG	E0→E7	E1→E8	E2→E9	E3→E10	E4→E11	AVG
Supervised	66.4	54.8	51.9	52.0	51.3	55.3	65.8	54.0	62.1	57.8	52.2	58.4	63.8	59.8	54.8	54.1	46.1	55.7
ClsTran	69.2	55.0	52.7	45.3	52.2	54.9	64.1	56.7	56.1	50.3	50.9	55.6	64.9	53.6	60.9	51.7	50.1	56.2
SimCLR	63.1	52.9	51.4	52.2	**59.7**	55.9	61.7	53.2	**64.2**	56.0	51.2	57.3	62.4	**60.2**	60.8	53.6	**57.1**	58.8
CPC	**71.8**	**61.9**	50.5	49.9	48.5	**56.5**	**68.6**	**62.9**	55.8	**60.3**	52.3	**60.0**	**72.6**	52.7	56.8	**59.6**	45.5	57.5
TS-TCC	63.0	51.8	**56.9**	**53.4**	52.2	55.5	65.6	53.5	58.4	54.6	**56.1**	57.6	71.2	58.2	**61.0**	57.5	51.4	**59.9**

	Intra-dataset Evaluation — With 1% of source domain labels																	
	E0→E7	E1→E8	E2→E9	E3→E10	E4→E11	AVG	E0→E7	E1→E8	E2→E9	E3→E10	E4→E11	AVG	E0→E7	E1→E8	E2→E9	E3→E10	E4→E11	AVG
Supervised	36.3	30.8	26.3	33.3	35.1	32.4	35.4	18.6	25.8	**33.2**	29.0	28.4	34.9	18.8	18.7	**36.2**	29.3	27.6
ClsTran	36.7	38.0	36.8	26.0	44.3	36.4	30.5	21.5	17.1	27.8	26.7	24.7	28.0	22.7	21.6	32.6	29.0	26.8
SimCLR	34.3	29.7	31.6	31.2	38.7	33.1	44.9	21.0	23.5	30.6	38.7	31.7	35.8	32.7	30.5	**36.2**	36.1	34.3
CPC	**56.3**	**45.6**	**47.9**	35.4	**49.2**	**46.9**	**50.5**	27.8	**41.3**	31.1	34.1	**37.0**	**57.4**	**42.9**	37.1	30.1	35.4	40.6
TS-TCC	51.1	44.1	40.8	**43.5**	39.3	43.8	40.2	**28.6**	34.7	29.4	**40.4**	34.6	54.9	37.4	**39.0**	35.7	**41.1**	**41.6**

	Inter-dataset Evaluation — With 100% of source domain labels																	
	S0→E7	S1→E8	S2→E9	S3→E10	S4→E11	AVG	S0→E7	S1→E8	S2→E9	S3→E10	S4→E11	AVG	S0→E7	S1→E8	S2→E9	S3→E10	S4→E11	AVG
Supervised	**58.0**	24.3	50.1	26.9	30.4	37.9	42.6	**44.6**	48.4	**40.7**	24.4	**40.1**	51.0	29.1	49.7	32.4	18.1	36.1
ClsTran	48.7	29.8	47.4	39.5	30.9	39.3	49.5	26.2	45.7	28.6	20.3	34.1	50.7	**36.8**	52.2	**42.4**	17.6	40.0
SimCLR	52.0	24.0	50.5	30.2	**32.3**	37.8	49.4	16.2	39.8	27.6	**31.9**	33.0	**53.0**	26.1	47.4	25.7	28.5	36.1
CPC	57.8	30.4	**52.4**	**45.0**	27.6	**42.7**	**55.1**	32.2	**50.0**	27.4	18.1	36.6	46.9	27.4	**56.4**	28.1	**30.1**	37.8
TS-TCC	54.1	**42.1**	52.3	43.1	20.1	42.3	54.2	30.7	42.8	35.7	19.5	36.6	51.9	36.6	53.0	31.1	29.1	**40.3**

	Inter-dataset Evaluation — With 1% of source domain labels																	
	S0→E7	S1→E8	S2→E9	S3→E10	S4→E11	AVG	S0→E7	S1→E8	S2→E9	S3→E10	S4→E11	AVG	S0→E7	S1→E8	S2→E9	S3→E10	S4→E11	AVG
Supervised	31.2	31.2	24.4	31.0	19.6	27.5	22.0	**41.4**	18.8	**36.1**	13.1	26.3	32.8	34.4	22.5	31.4	14.0	27.0
ClsTran	35.5	32.4	36.8	30.2	18.1	30.6	17.6	26.7	23.1	31.6	13.3	22.5	21.0	22.2	17.1	29.3	10.3	20.0
SimCLR	38.7	35.2	37.8	36.3	**20.7**	33.7	29.0	27.3	27.0	31.1	13.1	25.5	**48.6**	26.3	31.9	**38.1**	12.1	31.4
CPC	**44.5**	**45.5**	37.2	33.3	15.8	**35.3**	**42.7**	30.5	**35.2**	34.6	10.0	**30.6**	39.9	29.4	**34.6**	30.3	14.0	29.6
TS-TCC	35.4	41.7	**42.9**	**40.5**	15.7	35.2	34.4	31.1	28.7	29.2	**14.7**	27.6	30.1	**46.7**	33.4	36.3	**16.8**	**32.7**

E. Robustness to Domain-Shift

In some scenarios, we may afford to label (some) samples of one subject, and we aim to transfer the knowledge from this subject to another unlabeled and out-of-distribution subject. This distribution shift can be caused by a different data collection methodology or differences in subjects' health status. To deal with this challenging scenario, some recent works proposed transfer learning and unsupervised domain adaptation algorithms to mitigate the domain shift [38], [39], [40]. In this section, we investigate the transferability of supervised training against self-supervised pretraining under the domain-shift settings on five random cross-domain (cross-subject) scenarios. We explore different scenarios for this evaluation.

1) Intra-Dataset Evaluation: We chose the source and target subjects from the Sleep-EDF dataset to simulate the transferability across different subjects within the same data collection methodology. Nevertheless, this scenario still represents a domain shift since the subjects may have different health status.

We first assume access to the full source domain labels. Hence, in the 'Supervised' experiment, we train the model with the full source domain labels and test it directly on the target domain. For the SSL algorithms, we pretrain the model with the full *unlabeled* source domain data, and fine-tune the pretrained model with 100% of the source domain labels, then test on the target domain. Second, we repeat the above experiments with assuming access to only 1% of the source domain labels. The results of these two experiments are shown in the first two tables in Table IV.

Notably, with the availability of 100% of source domain labels, the advantage of SSL pretraining over supervised training is minor. We find that the best-performing SSL algorithm improves only 1.2, 1.6, and 4.2% in DeepSleepNet, AttnSleep, and 1D-CNN respectively. However, this advantage increases when only 1% of the source domain labels are available, where the improvement increases to become 14.5, 8.6, and 14% in the three SSC models. This shows the importance of self-supervised pretraining to boost transferability in the few-labels regime. We also notice a clear advantage of the robustness of contrastive SSL algorithms against the auxiliary SSL task.

2) Inter-Dataset Evaluation: Another scenario is to train the model on a source domain from a different dataset. The domain shift can be regarded to different collection tools or different health status of subjects. To simulate this scenario and remove the bias to a specific dataset, we conduct other experiments, where the source and target subjects are from different datasets. Specifically, we select the source subjects from the SHHS dataset, while the target subjects are from the Sleep-EDF dataset. To handle the inconsistent sampling rates across the two datasets, we down-sampled subjects from the

SHHS dataset to match the sampling rate of the Sleep-EDF dataset, i.e., 100 Hz.

Similar to the inter-domain evaluation, we experiment with 100% and 1% of the source domain labels, as shown in the third and fourth tables in Table IV. Despite the difficulty of the task, the self-supervised pretrained models still demonstrate superior transferability compared to supervised training techniques. This superiority over supervised training with 100% of source domain labels is present in DeepSleepNet and 1D-CNN models with an improvement of 4.8 and 4.2% respectively. However, with 1% of source domain labels, the SSL pretraining outperforms supervised transferability with a noticeable gain. Additionally, contrastive SSL algorithms are still more robust against the domain shift than the auxiliary method in cross-dataset scenarios.

VI. Discussion & Recommendations

In this paper, we studied whether self-supervised pretraining can help improve the performance of existing sleep stage classification models in the few-labeled data regime. Our experiments were held with four SSL algorithms and applied to three SSC models on three different datasets. The experimental results suggest the following conclusions.

- Contrastive SSL algorithms guarantee the superior performance of SSC models over supervised training in the few-labeled data settings.
- Contrastive SSL algorithms are robust against sleep data imbalance, and this imbalance does not affect the quality of learned representations.
- Self-supervised pretraining improves the out-of-domain transferability performance in SSC models.
- SSL with predictive tasks can improve the temporal learning capability of SSC models.

The above conclusions reveal some potential future works to enhance the SSL algorithms proposed for sleep stage classification. First, we find that the auxiliary task, ClsTran, yields lower performance even than the supervised training in most cases. Therefore, it is important to study the SSC problem and propose a new SSC-specific auxiliary task to be more beneficial to the downstream performance, similar to the proposed tasks in [41].

Second, our experiments included two contrastive SSL algorithms that rely on data augmentations to choose the positive and negative pairs, i.e., SimCLR and TS-TCC. These two methods consider *only* the augmented view of each same sample as the positive pair, and all the other samples are considered negative pairs. However, some of these negative pairs may share the same label and semantic information with that anchor sample, and pulling them away from each other may deteriorate the performance. Therefore, one way to improve these algorithms is to reduce the number of false negative samples when applying contrastive learning. In addition, designing well-suited augmentations for sleep EEG data can learn more effective representations.

Third, SSL algorithms showed limited improvement to the minority classes in the sleep data, i.e., N1 and N3, which limits the overall improvement. Therefore, another research direction is to study how can self-supervised algorithms learn more about the characteristics of minority classes during pretraining. Forth, we noticed that SSL algorithms had a limited transferability improvement, which can be further investigated. Last, based on our experiments ensemble multiple SSL algorithms do not always yield an improved performance (See Section SIII-D in the supplementary materials). Therefore, developing a novel methodology that can ensemble multiple self-supervised learning algorithms and achieves a relatively high-performance gain can be further explored.

VII. Conclusion

In this paper, we assess the efficacy of different self-supervised learning (SSL) algorithms to improve the performance of sleep stage classification (SSC) models under the few-labels settings. The experimental results reveal that contrastive SSL algorithms can learn more robust and invariant representations about the sleep EEG data. In addition, SSL algorithms that include predictive tasks can learn temporal features about EEG data during pretraining, and hence lessens the need for a temporal encoder in the SSC models. Moreover, self-supervised pretraining can improve the robustness of SSC models against data imbalance and domain shift problems. Hence, we recommend pretraining existing SSC models with contrastive SSL algorithms to become more practical in real-world label-scarce scenarios.

References

[1] I. Perez-Pozuelo et al., "The future of sleep health: A data-driven revolution in sleep science and medicine," *npj Digit. Med.*, vol. 3, p. 42, Mar. 2020.

[2] R. B. Berry et al., "AASM scoring manual updates for 2017 (version 2.4)," *J. Clin. Sleep Med.*, vol. 13, no. 5, pp. 665–666, 2017.

[3] P. Jadhav and S. Mukhopadhyay, "Automated sleep stage scoring using time-frequency spectra convolution neural network," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–9, 2022.

[4] J. Phyo, W. Ko, E. Jeon, and H.-I. Suk, "Enhancing contextual encoding with stage-confusion and stage-transition estimation for EEG-based sleep staging," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 1301–1305.

[5] H. Phan, K. Mikkelsen, O. Y. Chen, P. Koch, A. Mertins, and M. De Vos, "SleepTransformer: Automatic sleep staging with interpretability and uncertainty quantification," *IEEE Trans. Biomed. Eng.*, vol. 69, no. 8, pp. 2456–2467, Aug. 2022.

[6] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 1, 2020, pp. 1597–1607.

[7] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.

[8] X. Chen and K. He, "Exploring simple Siamese representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15750–15758.

[9] J.-B. Grill et al., "Bootstrap your own latent—A new approach to self-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 21271–21284.

[10] E. Eldele et al., "Time-series representation learning via temporal and contextual contrasting," in *Proc. 13th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2021, pp. 2352–2359.

[11] J. Ye, Q. Xiao, J. Wang, H. Zhang, J. Deng, and Y. Lin, "CoSleep: A multi-view representation learning framework for self-supervised learning of sleep stage classification," *IEEE Signal Process. Lett.*, vol. 29, pp. 189–193, 2022.

[12] 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.

[13] E. Eldele et al., “An attention-based deep learning approach for sleep stage classification with single-channel EEG,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 29, pp. 809–818, 2021.

[14] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” 2018, *arXiv:1807.03748*.

[15] A. I. Humayun, A. S. Sushmit, T. Hasan, and M. I. H. Bhuiyan, “End-to-end sleep staging with raw single channel EEG using deep residual ConvNets,” in *Proc. IEEE EMBS Int. Conf. Biomed. Health Informat. (BHI)*, May 2019, pp. 1–5.

[16] 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, Aug. 2020, Art. no. 102037.

[17] Q. Cai, Z. Gao, J. An, S. Gao, and C. Grebogi, “A graph-temporal fused dual-input convolutional neural network for detecting sleep stages from EEG signals,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 2, pp. 777–781, Feb. 2021.

[18] H. Phan, F. Andreotti, N. Cooray, Y. Oliver Chen, and M. D. 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.

[19] 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.

[20] M. Noroozi and P. Favaro, “Unsupervised learning of visual representations by solving jigsaw puzzles,” in *Computer Vision–(ECCV)*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 69–84.

[21] S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised representation learning by predicting image rotations,” in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–16.

[22] M. N. Mohsenvand, M. R. Izadi, and P. Maes, “Contrastive representation learning for electroencephalogram classification,” in *Proc. Mach. Learn. Health NeurIPS Workshop*, 2020, pp. 238–253.

[23] X. Jiang, J. Zhao, B. Du, and Z. Yuan, “Self-supervised contrastive learning for EEG-based sleep staging,” in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–8.

[24] H. Banville, O. Chehab, A. Hyvärinen, D.-A. Engemann, and A. Gramfort, “Uncovering the structure of clinical EEG signals with self-supervised learning,” *J. Neural Eng.*, vol. 18, no. 4, Aug. 2021, Art. no. 046020.

[25] Q. Xiao et al., “Self-supervised learning for sleep stage classification with predictive and discriminative contrastive coding,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 1290–1294.

[26] H. Lee, E. Seong, and D.-K. Chae, “Self-supervised learning with attention-based latent signal augmentation for sleep staging with limited labeled data,” in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 3868–3876.

[27] H. Zhang, J. Wang, J. Xiong, Y. Ding, Z. Gan, and Y. Lin, “Expert knowledge inspired contrastive learning for sleep staging,” in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2022, pp. 1–6.

[28] A. Saeed, T. Ozcelebi, and J. Lukkien, “Multi-task self-supervised learning for human activity detection,” *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 3, no. 2, pp. 1–30, Jun. 2019.

[29] P. Sarkar and A. Etemad, “Self-supervised ECG representation learning for emotion recognition,” *IEEE Trans. Affect. Comput.*, vol. 13, no. 3, pp. 1541–1554, Jul. 2022.

[30] E. Eldele et al., “Self-supervised contrastive representation learning for semi-supervised time-series classification,” 2022, *arXiv:2208.06616*.

[31] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, “A survey on contrastive self-supervised learning,” *Technologies*, vol. 9, no. 1, p. 2, Dec. 2020.

[32] 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.

[33] G.-Q. Zhang et al., “The national sleep research resource: Towards a sleep data commons,” *J. Amer. Med. Inform. Assoc.*, vol. 25, no. 10, pp. 1351–1358, 2018.

[34] S. F. Quan et al., “The sleep heart health study: Design, rationale, and methods,” *Sleep*, vol. 20, no. 12, pp. 1077–1085, 1997.

[35] S. Khalighi, T. Sousa, J. M. Santos, and U. Nunes, “ISRUC-Sleep: A comprehensive public dataset for sleep researchers,” *Comput. Methods Programs Biomed.*, vol. 124, pp. 180–192, Feb. 2016.

[36] A. Newell and J. Deng, “How useful is self-supervised pretraining for visual tasks?” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7343–7352.

[37] H. Liu, Z. Jeff HaoChen, A. Gaidon, and T. Ma, “Self-supervised learning is more robust to dataset imbalance,” in *Proc. Int. Conf. Learn. Represent.*, 2022, pp. 1–24.

[38] E. Eldele et al., “ADAST: Attentive cross-domain EEG-based sleep staging framework with iterative self-training,” *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 1, pp. 210–221, Feb. 2023.

[39] 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.

[40] M. Ragab, E. Eldele, Z. Chen, M. Wu, C.-K. Kwoh, and X. Li, “Self-supervised autoregressive domain adaptation for time series data,” *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 23, 2022, doi: 10.1109/TNNLS.2022.3183252.

[41] N. Wagh et al., “Domain-guided self-supervision of eeg data improves downstream classification performance and generalizability,” in *Proc. Mach. Learn. Health*, vol. 158, Dec. 2021, pp. 130–142.