Self-supervised Electroencephalogram Representation Learning for Automatic Sleep Staging

Chaoqi Yang[1], Danica Xiao[2], M. Brandon Westover[3,4], Jimeng Sun[1*]

[1]University of Illinois Urbana-Champaign, [2]Amplitude, [3]Massacussetts General Hospital, [4]Havard Medical School

Correspondence to Jimeng Sun, PhD, 201 North Goodwin Avenue Urbana, IL 61801; Email: jimeng@illinois.edu

ABSTRACT

Objective: In this paper we aim to learn robust vector representations from massive unlabeled Electroencephalogram (EEG) signals, such that the learned representations (1) are expressive enough to replace the raw signals in the sleep staging task; and (2) provide better predictive performance than supervised models in scenarios of fewer labels and noisy samples.

Materials and Methods: We propose a self-supervised model, named *Contrast with the World Representation* (ContraWR), for EEG signal representation learning, which uses global statistics from the dataset to distinguish signals associated with different sleep stages. The ContraWR model is evaluated on three real-world EEG datasets that include both at-home and in-lab recording settings.

Results: ContraWR outperforms recent self-supervised learning methods, MoCo, SimCLR, BYOL, SimSiam on the sleep staging task across three datasets. ContraWR also beats supervised learning when fewer training labels are available (e.g., 4% accuracy improvement when less than 2% data is labeled). Moreover, the model provides informative representations in 2D projection.

Discussion: The proposed model can be generalized to other unsupervised physiological signal learning tasks. Future directions include exploring task-specific data augmentations and combining self-supervised with supervised methods, building upon the initial success of self-supervised learning in this paper.

Conclusions: We show that ContraWR is robust to noise and can provide high-quality EEG representations for downstream prediction tasks. In low-label scenarios (e.g., only 2% data has labels), ContraWR shows much better predictive power (e.g., 4% improvement on sleep staging accuracy) than supervised baselines.

GitHub Code Repository: https://github.com/ycq091044/ContraWR

INTRODUCTION

Deep learning models have shown great success in automating tasks in sleep medicine by learning from large amounts of labeled EEG data[1]. EEG data are collected from patients wearing clinical sensors, which generate real-time multi-modal signal data. A common challenge in classifying physiological signals, including EEG, is the lack of sufficient numbers of high-quality labels. This paper proposes a new self-supervised model which exploits latent structure in massive unlabeled and noisy data and provides robust feature representations for downstream classification tasks such as sleep staging.

Self-supervised learning (also known as contrastive learning) aims at learning a feature encoder which maps input signals into a vector representation using unlabeled data. Self-supervised methods involve two steps: (i) a **pretext task**: to learn the encoder without labels; (ii) a **supervised task**: to evaluate the learned encoder with a small amount of labeled data. During the pretext task, recent methods (e.g., MoCo[2], SimCLR[3]) use the encoder to construct *positive* and *negative pairs* from the unlabeled data and then optimize the encoder by pushing positive pairs closer and negative pairs farther away. A positive pair consists of two different augmented versions of the same sample, while a negative pair is generated from two different samples. For example, the augmentation method for EEG data can be denoising or channel flipping. Existing negative sampling strategies often incur sampling bias[4,5], especially for noisy EEG data, which significantly hurts performance[6].

Technically, this paper contributes to the pretext task, where we address the inherent limitations of negative sampling in the existing self-supervised methods (e.g., MoCo[2], SimCLR[3]) by leveraging global data statistics. In contrastive learning, positive pairs bring similarity information, while negative pairs provide contrastive information. We propose a new method to address the limitation in negative sampling, named *contrast with the world representation (ContraWR)*, where an average representation over the dataset (called the *world*

representation) is presented as the only contrastive information. We propose robust contrastive guidance under the absence of labels: *the representation similarity between positive pairs is stronger than the similarity to the world representation*. Derived from global data statistics, the world representation brings robust contrastive information even in noisy environments. Moreover, we strengthen our model with an instance-aware world representation for each sample, where closer samples are assigned larger weights. Our experiments show that the instance-aware world representation makes the model more accurate.

We evaluate ContraWR on the sleep staging task with three real-world EEG datasets. Our model achieves results comparable to or better than recent self-supervised methods, MoCo [7], SimCLR[3], BYOL[8] and SimSiam[9]. The results also show that contrastive methods, especially our ContraWR method, are much more powerful in low-label scenarios than supervised learning (e.g., 4% accuracy improvement on sleep staging with less than 2% training data of Sleep EDF dataset).

BACKGROUND AND RELATED WORKS

Self-supervised Learning

Many deep generative methods have been proposed for unsupervised representation learning. They mostly rely on auto-encoding[10-12] or adversarial training[13-15]. Mutual information (MI) maximization is also popular for compressing input data into a latent representation[16-18].

Recently, self-supervised contrastive learning[3,7] has become popular, where loss functions are devised from representation similarity and negative sampling. However, one recent publication[4] highlighted inherent limitations of negative sampling and showed that it hurts the learned representation significantly[5]. To address these limitations, Chuang et al.[5] approximated the per-class negative sample distribution using the overall distribution and per-class distribution. However, without label information, the true class distribution is unknown. Grill et al.[8] and Chen et al.[9] proposed ignoring negative samples and learning latent representations using only positive pairs.

In this paper we replace negative sampling with the global average (i.e., the world representation). We argue and provide experiments showing that that contrasting with the world representation is more powerful and robust.

EEG Sleep Staging

Before the emergence of deep learning, several traditional machine learning approaches[19-22] significantly advanced the field using hand-crafted features, as highlighted in [23]. Recently, deep learning models have been applied to various large sleep databases. SLEEPNET[23] built a comprehensive system combining many machine learning models to learn sleep signal representations. Biswal et al.[1] designed a multi-layer RCNN model to process multi-channel signals from EEG. To provide interpretable stage prototypes, Al-Hussaini et al.[24] developed a SLEEPER model that utilizes a particular deep learning approach called prototype learning guided by a decision tree to provide more interpretable results. These works rely on a large labeled training set. In this paper, we learn the signal representations from mostly unlabeled data, which is more challenging.

Self-supervised Learning on Physiological Signals

While image[25,26], video[27], language[28,29] and speech[30] representations have benefited from contrastive learning, research on learning physiological signals has been limited[31,32]. Lemkhenter et al.[33] proposed phase and amplitude coupling for physiological data augmentation. Banville et al.[2] conducted representation learning on EEG signals, targeted toward monitoring and pathology screening tasks, without utilizing frequency information. Cheng et al.[34] learned subject-aware representations for ECG data and tested various augmentation methods. While most of these methods are based on *pairwise* similarity comparison, our model brings the contrastive information from *global* data statistics, providing more robust representations. Also, we extract signal information from spectral domain.

MATERIALS AND METHODS

Problem Formulation

The input features of each subject's EEG recording are multi-channel brain waves. First, subjects are classified into pretext/training/test groups. The training and test groups are small, but their data are labeled, while the pretext group consists of a large number of unlabeled recordings. Within each group, recordings are

segmented into disjoint 30-second periods, where each period is called an *epoch*, denoted $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{N}}$. Each epoch has the same format: $\boldsymbol{C}$ input channels and $\boldsymbol{N}$ samples from each channel.

The EEG representation learning problem requires building an encoder $\boldsymbol{f}(\cdot)$ from the pretext group (without labeling information), which maps one epoch $\boldsymbol{x}$ into a vector representation $\boldsymbol{h} \in \mathbb{R}^{\boldsymbol{d}}$, where $\boldsymbol{d}$ is the latent dimensionality, such that the representation $\boldsymbol{h}$ can replace raw signal for downstream tasks. The evaluation of the encoder $\boldsymbol{f}(\cdot)$ is conducted on the training and test data. We focus on sleep staging as the downstream task, where the sample $\boldsymbol{x}$ from each 30-second epoch is categorized into five stages $\boldsymbol{W}, \boldsymbol{R}, \boldsymbol{N1}, \boldsymbol{N2}, \boldsymbol{N3}$, based on American Academy of Sleep Medicine (AASM) scoring standards[35].

Background and Concepts

Self-supervised learning uses representation similarity to discriminate unlabeled signals, with an *encoder network* $f(\cdot)\colon \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{N}} \to \mathbb{R}^{\boldsymbol{d}}$ and a nonlinear *projection function* $g(\cdot)\colon \mathbb{R}^{\boldsymbol{d}} \to \mathbb{R}^{\boldsymbol{m}}$. For a given signal $\boldsymbol{x}$ from the pretext group, data augmentation methods $a(\cdot)$[1] first produce two different augmented signals $\tilde{\boldsymbol{x}}_{\boldsymbol{i}}, \tilde{\boldsymbol{x}}_{\boldsymbol{j}}$ during each iteration, which are then transformed into $\boldsymbol{h}_{\boldsymbol{i}}, \boldsymbol{h}_{\boldsymbol{j}} \in \mathbb{R}^{\boldsymbol{d}}$ by $f(\cdot)$ and further into $\boldsymbol{z}_{\boldsymbol{i}}, \boldsymbol{z}_{\boldsymbol{j}} \in \mathbb{R}^{\boldsymbol{m}}$ by $g(\cdot)$. The vectors $\boldsymbol{z}_{\boldsymbol{i}}, \boldsymbol{z}_{\boldsymbol{j}}$ are finally normalized with the $L2$ norm onto the unit hypersphere, $\frac{\boldsymbol{z}}{||\boldsymbol{z}||} \in \mathbb{S}^{m-1}$. We call $\frac{\boldsymbol{z}_{\boldsymbol{i}}}{||\boldsymbol{z}_{\boldsymbol{i}}||}$ the *anchor*, $\frac{\boldsymbol{z}_{\boldsymbol{j}}}{||\boldsymbol{z}_{\boldsymbol{j}}||}$ the *positive sample,* and these two together are called a *positive pair*. For a large number of projections $\{\boldsymbol{z}_{\boldsymbol{k}} \in \mathbb{R}^{m}\}$ derived from other randomly selected signals (by negative sampling), their representation $\left\{\frac{\boldsymbol{z}_{\boldsymbol{k}}}{||\boldsymbol{z}_{\boldsymbol{k}}||} \in \mathbb{S}^{m-1}\right\}$ is commonly conceived of as *negative samples* (though they are random samples), and any one of them together with the anchor is called a *negative pair*. The loss functions ($\boldsymbol{L}$) is derived from the similarity comparison between positive pair and negative pairs (e.g., encouraging similarity of positive pairs to be stronger than that of the negative pairs, referred to as the noise contrastive estimation (NCE) loss[36] in Appendix). A conceptual flow is illustrated as,

$$\boldsymbol{x} \Rightarrow_{a(\cdot)} \tilde{\boldsymbol{x}} \Rightarrow_{f(\cdot)} \boldsymbol{h} \Rightarrow_{g(\cdot)} \boldsymbol{z} \Rightarrow_{L2} \frac{\boldsymbol{z}}{||\boldsymbol{z}||} \Rightarrow_{def} \boldsymbol{L}\,.$$

[1] This paper uses bandpass filtering, noising, channel flipping, and rotation; see Figure 1. We conduct ablation studies on the augmentation methods in experiment and provide the implementation details in the Appendix. Note that, the Augmentation methods do not change the dimensions of the signal.

To reduce clutter, we use $\boldsymbol{z}$ to denote the $L2$ normalized version in the rest of the paper.

(I). ContraWR: Contrast with the World Representation

As mentioned above, negative sampling can introduce bias for the pretext task and can undermine representation quality. We propose a new self-supervised learning method, *Contrast with the World Representation (ContraWR)*. ContraWR replaces the large number of negative samples with a single average representation over the dataset, called the *world representation*. The world representation works as a reference to calibrate the model, making the pretext task more robust. Our loss function in pretext task follows the principle: *the representation similarity between a positive pair should be stronger than the similarity between the anchor and the world representation*.

The world representation. Assume $\boldsymbol{z_i}$ is the anchor, $\boldsymbol{z_j}$ is the positive sample, and $\boldsymbol{z_k}$ is a random sample. We generate an average representation of the dataset, $\boldsymbol{z_w}$ as the only contrastive information. To formalize, we assume $p(\cdot)$ is the sample distribution over the dataset, independent of the anchor $\boldsymbol{z_i}$. The world representation $\boldsymbol{z_w}$ is defined by[2],

$$\boldsymbol{z_w} = E_{k\sim p(\cdot)}[\boldsymbol{z_k}].$$

Here, we denote $\boldsymbol{D} = \{\boldsymbol{z}: ||\boldsymbol{z}|| \leq 1, \boldsymbol{z} \in \mathbb{R}^m\}$. Obviously, $\boldsymbol{z_w} \in \boldsymbol{D}$.

Gaussian kernel measure. We adopt a Gaussian kernel defined on $\boldsymbol{D}$, $sim(\boldsymbol{x}, \boldsymbol{y})$: $\boldsymbol{D} \times \boldsymbol{D} \rightarrow (0,1]$ as a similarity measure. Formally, given two projections $\boldsymbol{z_i}, \boldsymbol{z_j}$ the similarity is defined as,

$$sim(\boldsymbol{z_i}, \boldsymbol{z_j}) = exp\left(-\frac{||\boldsymbol{z_i}-\boldsymbol{z_j}||^2}{2\sigma^2}\right),$$

where σ is a hyperparameter.

Loss function. For the anchor $\boldsymbol{z_i}$, the positive sample $\boldsymbol{z_j}$ and the world representation $\boldsymbol{z_w}$, we devise a triplet loss,

[2] In the experiment, $\boldsymbol{z_w}$ is approximated by Monte Carlo method within each batch, i.e., we use the average value over the batch, $\boldsymbol{z_w} = \frac{1}{M}\sum_{i=1}^{M}\boldsymbol{z_j}$, where M is the batch size.

$$\boldsymbol{L}(i,j) = \left[sim(\boldsymbol{z_i}, \boldsymbol{z_w}) + \delta - sim(\boldsymbol{z_i}, \boldsymbol{z_j})\right]_+,$$

Where $\delta > 0$ is the empirical margin. The loss is minimized over batches, ensuring that the similarity of positive pair, $sim(\boldsymbol{z_i}, \boldsymbol{z_j})$, is larger than the similarity to the world representation, $sim(\boldsymbol{z_i}, \boldsymbol{z_w})$, by a margin of δ.

The pipeline of our ContraWR is shown in Figure 2. The online networks $f_\theta(\cdot), g_\theta(\cdot)$ and the momentum target networks $f_\phi(\cdot), g_\phi(\cdot)$ share an identical network structure. Encoder networks $f_\theta(\cdot), f_\phi(\cdot)$ map two augmented versions of the same signal to latent representations, respectively. Then, the projection functions $g_\theta(\cdot), g_\phi(\cdot)$ project the latent representations onto a unit hypersphere, where the loss is defined. During optimization the online networks are updated by gradient descent, and the target networks update parameters from the online network with an exponential moving average (EMA) trick[7].

$$\theta^{(n+1)} \leftarrow \theta^{(n)} - \eta \cdot \nabla_\theta \boldsymbol{L}$$

$$\phi^{(n+1)} \leftarrow \lambda \cdot \phi^{(n)} + (1-\lambda) \cdot \theta^{(n+1)}$$

where n indicates the n-th update, η is the learning rate, and λ is a weight hyperparameter. After this optimization on the **pretext task**, the encoder network $f_\theta(\cdot)$ is ready to be evaluated on the training and test group for the **supervised task**.

(II). ContraWR+: Contrast with Instance-aware World Representation

To learn a better representation, we introduce a weighted averaged world representation, based on the harder principle: *the similarity between a positive pair should be stronger than the similarity between the anchor and the weighted average of the world/dataset, where the weight is set higher for closer samples.* This is a more difficult objective than the simple global average in ContraWR.

Instance-aware world representation. The world representation is enhanced by modifying the sampling distribution to be instance-specific. We define $p(\cdot\,|\boldsymbol{z})$ as the instance-aware sampling distribution of an anchor $\boldsymbol{z}$, by contrast with the global sample distribution $p(\cdot)$,

$$p(\cdot\,|\boldsymbol{z}) \propto \exp\left(\frac{\langle \cdot, \boldsymbol{z} \rangle}{T}\right),$$

where $T > 0$ is a temperature hyperparameter, such that similar samples are selected with higher probability parametrized by $p(\cdot\,|\boldsymbol{z})$. Accordingly, for an anchor $\boldsymbol{z}_i$, the instance-aware world representation becomes,

$$\boldsymbol{z}_{\boldsymbol{w}(\boldsymbol{i})} = E_{k\sim p(\,\cdot|\boldsymbol{z}_i\,)}[\boldsymbol{z}_k] = \frac{E_{k\sim p}\left[\exp\left(\frac{\langle \boldsymbol{z}_k, \boldsymbol{z}_i\rangle}{T}\right)\cdot \boldsymbol{z}_k\right]}{E_{k\sim p}\left[\exp\left(\frac{\langle \boldsymbol{z}_k, \boldsymbol{z}_i\rangle}{T}\right)\right]}.$$

Here, T controls the contrastive hardness of the world representation. When $T \to \infty$, $p(\cdot\,|\boldsymbol{z})$ is asymptotically identical to $p(\cdot)$, and the above equation reduces to the simple global average; while $T \to 0^+$, the form becomes trivial, $\boldsymbol{z}_{\boldsymbol{w}(\boldsymbol{i})} = argmax_{\boldsymbol{z}_k}(sim(\boldsymbol{z}_i, \boldsymbol{z}_k))$. We have tested different T and find the model is not sensitive to T over a wide range. Here, $\boldsymbol{z}_{\boldsymbol{w}(\boldsymbol{i})}$ is also approximated Monte Carlo sampling. We can re-write the similarity measure given the anchor $\boldsymbol{z}_i$ and the new world representation $\boldsymbol{z}_{\boldsymbol{w}(\boldsymbol{i})}$ as:

$$sim\left(\boldsymbol{z}_i, \boldsymbol{z}_{\boldsymbol{w}(\boldsymbol{i})}\right) = sim\left(\boldsymbol{z}_i, E_{k\sim p(\cdot|\boldsymbol{z}_i)}[\boldsymbol{z}_k]\right) = \exp\left(-\frac{1}{2\sigma^2}||\boldsymbol{z}_i - \frac{E_{k\sim p}\left[\exp\left(\frac{\langle \boldsymbol{z}_k, \boldsymbol{z}_i\rangle}{T}\right)\cdot \boldsymbol{z}_k\right]}{E_{k\sim p}\left[\exp\left(\frac{\langle \boldsymbol{z}_k, \boldsymbol{z}_i\rangle}{T}\right)\right]}||^2\right).$$

Later, we also update the triplet loss with this new similarity measure.

EXPERIMENTS

EEG Datasets

To evaluate the proposed method, we consider three real-world EEG datasets:

- *Sleep Heart Health Study (SHHS)*[37,38] is a multi-center cohort study from the National Heart Lung & Blood Institute assembled to study sleep-disordered breathing, which contains 5,445 recordings. Each recording has 14 Polysomnography (PSG) channels, and the recording frequency is 125.0 Hz. We use the C3/A2 and C4/A1 EEG channels.
- *Sleep EDF*[39] is another benchmark dataset collected in a 1987-1991 study of age effects on sleep in healthy Caucasians aged 25-101 who were taking non sleep-related medications, which contains 153 full-night EEG recordings with recording frequency 100.0 Hz. We extract the Fpz-Cz/Pz-Oz EEG channels as the raw inputs to the model. The first two datasets are all at-home PSG recordings.
- *MGH Sleep*[1] is collected from sleep laboratory at Massachusetts General Hospital (MGH), where six EEG channels (i.e., F3-M2, F4-M1, C3-M2, C4-M1, O1-M2, O2-M1) are used for sleep staging recorded at 200.0

Hz frequency. After filtering out mismatched signals and missing labels, we finally get 6,478 recordings. Dataset statistics can be found in Table 1, and class label distribution is in Table 2.

For these datasets, the ground truth labels were released by the original data publishers. For each dataset, subjects are randomly assigned to pretext group, training group and test group with different proportions (90%: 5%: 5% for Sleep EDF and MGH, 98%: 1%: 1% for SHHS, since they have different amount of data). We further segment each subject's recording into non-overlapping 30-second epochs, which are the data samples in the experiment. All epochs segmented from one subject are placed within the same group. The pretext group is used for self-supervised learning, so we remove sleep stage labels from this group. Training and test groups have labels. The training group is used to learn the predictive model either on raw EEG signals (for supervised methods) or on top of the learned feature encoder (for self-supervised methods). The test group is used to evaluate performance.

Experimental setup

We include several recent self-supervised methods for comparison,

- **MoCo**[7] devises two parallel encoders with exponential moving average (EMA). It also utilizes a frequently updated memory table to store new negative samples.
- **SimCLR**[3] uses one encoder network to generate both anchor and positive samples, where negative samples are collected from the same batch.
- **BYOL**[8] also employs two encoders: one online network and one target network. They put one more predictive layer on top of the online network to predict (reconstruct) the result from the target network, while no negative samples are presented.
- **SimSiam**[9] is the simplest self-supervised architecture with (i) one encoder network and (ii) no negative samples.

Signal Augmentation. For the experiments, we use four augmentation methods, shown in Figure 1: (i) Bandpass Filtering. To reduce noise, we use an order-1 Butterworth filter (the bandpass is specified in appendix). (ii) Noising. We add extra high-frequency or low-frequency noise to each channel. (iii) Channel Flipping. Corresponding sensors from the left side and the right of the head are swapped. (iv) Shifting. Within one sample,

we advance or delay the signal for a certain time span. Detailed configurations of augmentation methods vary for the three datasets, and we list them in the Appendix.

The encoder network architecture is shown in Figure 3. For a fair comparison, these baseline approaches use the same augmentation and encoder architecture. We also consider a supervised counterpart (called **Supervised**) with the same encoder, on top of which we further add a 2-layer fully connected network (128-unit for Sleep EDF, 256-unit for SHHS, and 192-unit for MGH) for the classification task. The supervised model does not use the pretext group but is trained from scratch on raw EEG signals in the training group and tested on the test group. We also include an **Untrained Encoder** model as a baseline, where the encoder is not optimized in the pretext task. The self-supervised models go through the following evaluation procedure [3,7].

Evaluation Design. The encoder for self-supervised methods is pre-trained with pretext groups (without labels), and evaluated following the standard strategy[7,40]: training a separate logistic regression model (on top of the encoder) on data from the training group (during which the encoder is frozen) and test on new recordings. We evaluate performance on the sleep staging task with overall five-class classification *accuracy*. Each experiment is conducted with five different random seeds. For self-supervised methods, we optimize the encoder for 100 epochs (here, "epoch" is a concept in deep learning) with unlabeled data and use training/test group data for evaluation. For the supervised method, we train the model for 100 epochs. Our setting ensures the convergence of all models.

Evaluation Results

Performance Comparison. Comparisons on the downstream sleep staging task are shown in Table 3. All self-supervised methods outperform the Untrained Encoder model, indicating that the pretext task does produce useful features from unlabeled data. We observe that ContraWR and ContraWR+ both outperform the supervised model, suggesting that the feature representation provided by the encoder have better expressive power than the raw signals for the sleep staging task, at least with when the amount of labeled data available is not sufficient (e.g., less than 2% in Sleep EDF). Compared to other self-supervised methods, our proposed model also provides better predictive accuracy, i.e., about 1.3% on Sleep EDF, 0.8% on SHHS, 1.3% on MGH Sleep. MGH Sleep data contains more noise than the other two datasets (reflected by the relatively low accuracy with

supervised model on raw signals). It is notable that the performance gain is much more significant on MGH over other self-supervised or supervised models (about 3.3% relative improvement on accuracy) which suggests that the proposed models handle noisy environments better.

Evaluation on Augmentations. We also inspect the effectiveness of different augmentation methods for EEG signals, shown in Table 4. We empirically test all possible combinations of four considered augmentations: channel flipping, bandpass filtering, noising, rotation (illustrated in Figure 1). Since channel flipping cannot be applied solely, we combine it with other augmentations. The evaluation is conducted on Sleep EDF with ContraWR+ model. To sum up, all augmentation methods are beneficial, and together, they can further boost the classification performance.

Varying Amount of Training Data. To further investigate the benefits of self-supervised learning, we evaluate the effectiveness of learned representations with varying training data on Sleep EDF in Figure 4. The default setting is to split all the data into pretext/training/test groups by 90%: 5%: 5% (as in the experiments above). In this section, we keep the 5% test group unchanged and re-split the pretext and training groups, such that the training proportion becomes 0.5%, 1%, 2%, 5%, 10%, and the rest is used for the pretext group. This "re-splitting" is conducted at the subject level, after which we again segment each subject's recording within the group. We compare our ContraWR+ to MoCo, SimCLR, BYOL, SimSiam, and the supervised counterpart. Our model outperforms the baselines consistently with different amount of training data. For example, our model achieves similar performance (with only 5% data as training) compared to the best baseline, BYOL, which needs twice amount of training data (10% data as training). Also, compared to the supervised model, the self-supervised methods perform better when the labels are insufficient, e.g., only $\leq 2\%$ of the data are labeled.

Representation Projection. We next sought to assess the quality of the representation learned by ContraWR+ qualitatively. To do this, we use the codes/representations produced by ContraWR+ on the MGH dataset and randomly select 5,000 signal epochs per class from the dataset. The encoder is optimized on the pretext task, which is agnostic to stage labels. Therefore, the selected epochs are not necessarily from the test group. We extract feature representations for each sample through the encoder network and use uniform manifold approximation and projection (UMAP)[41] for 2D mapping. We finally color code samples according to sleep stage labels for illustration.

The 2D mapping is shown in Figure 5. We also compute the confusion matrix from the evaluation stage (based on the test group), also shown in Figure 5. In the UMAP projection, epochs from the same latent class are closely co-located, which means the pretext task extracts important information for sleep stage classification from the raw unlabeled EEG signals. Stage N1 overlaps with stages W, N2, and N3, which is as expected given that N1 is often ambiguous and thus difficult to classify even for trained experts [1].

Hyperparameter Ablation Study. To investigate the sensitivity of our model to hyperparameter settings, we test with different batch sizes and train on different values for the Gaussian parameter σ, temperature T, and margin δ. We focus on the ContraWR+ model and evaluate it on the Sleep EDF dataset. During the experiment, the default settings are batch size = 256, $\sigma = 2, T = 2, \delta = 0.2$, learning rate η = 2e-4, weight decay = 1e-4, epoch = 100. When testing on one hyperparameter, others are held fixed. Ablation study results are in shown in Figure 6; the red star indicates the default configuration. We see that the model is not sensitive to batch size. We see that over a large range (< 10) the model is insensitive to the Gaussian width σ. For temperature T, we noted previously that a very small T may be problematic, and a very large T reduces ContraWR+ to ContraWR. Based on the ablation experiments the performance is relatively insensitive to choices of T. For the margin δ, the distance difference is bounded (given fixed $\sigma = 2$),

$$||sim(\boldsymbol{z_i}, \boldsymbol{z_w}) - sim(\boldsymbol{z_i}, \boldsymbol{z_j})|| \leq ||\exp\left(-\frac{0^2}{2\sigma^2}\right) - \exp\left(-\frac{2^2}{2\sigma^2}\right)||^2 \approx 0.3935.$$

Thus, δ should be chosen large enough, i.e., $\delta \geq 0.1$.

CONCLUSION AND DISCUSSIONS

This paper is motivated by the need to learn effective EEG representations from large unlabeled noisy EEG datasets. We propose a self-supervised contrastive method, ContraWR, and its enhanced variant, ContraWR+. Instead of creating a large number of negative samples our method contrasts samples with an average representation of many samples. The model is evaluated on a downstream sleep staging task with three real-world EEG datasets. Extensive experiments show that the model is more powerful and robust than multiple baselines including MoCo, SimCLR, BYOL, and SimSiam. ContraWR+ also outperforms the supervised counterpart in label-insufficient scenarios.

DATA AVAILABILITY

The SHHS and Sleep EDF datasets are public sources and are available at https://sleepdata.org/datasets/shhs and https://physionet.org/content/sleep-edfx/1.0.0/. The MGH sleep data is provided by Massachusetts General Hospital (MGH) and may be shared on reasonable request to mwestover@mgh.harvard.edu. All the code is available at https://github.com/ycq091044/ContraWR.

REFERENCES

1 Biswal, S. *et al.* Expert-level sleep scoring with deep neural networks. *Journal of the American Medical Informatics Association* **25**, 1643-1650 (2018).

2 Banville, H., Chehab, O., Hyvarinen, A., Engemann, D. & Gramfort, A. Uncovering the structure of clinical EEG signals with self-supervised learning. *Journal of Neural Engineering* (2020).

3 Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709* (2020).

4 Arora, S., Khandeparkar, H., Khodak, M., Plevrakis, O. & Saunshi, N. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229* (2019).

5 Chuang, C.-Y., Robinson, J., Lin, Y.-C., Torralba, A. & Jegelka, S. Debiased contrastive learning. *Advances in Neural Information Processing Systems* **33** (2020).

6 Robinson, J., Chuang, C.-Y., Sra, S. & Jegelka, S. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592* (2020).

7 He, K., Fan, H., Wu, Y., Xie, S. & Girshick, R. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 9729-9738.

8 Grill, J.-B. *et al.* Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems* **33** (2020).

9 Chen, X. & He, K. Exploring Simple Siamese Representation Learning. *arXiv preprint arXiv:2011.10566* (2020).

10 Vincent, P. *et al.* Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research* **11** (2010).

11 Kingma, D. P. & Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

12 Baldi, P. in *Proceedings of ICML workshop on unsupervised and transfer learning.* 37-49.

13 Donahue, J., Krähenbühl, P. & Darrell, T. Adversarial feature learning. *arXiv preprint arXiv:1605.09782* (2016).

14 Goodfellow, I. *et al.* in *Advances in neural information processing systems.* 2672-2680.

15 Shrivastava, A. *et al.* in *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2107-2116.

16 Hjelm, R. D. *et al.* Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670* (2018).

17 Tschannen, M., Djolonga, J., Rubenstein, P. K., Gelly, S. & Lucic, M. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625* (2019).

18 Bachman, P., Hjelm, R. D. & Buchwalter, W. in *Advances in Neural Information Processing Systems.* 15535-15545.

19 Fraiwan, L., Lweesy, K., Khasawneh, N., Fraiwan, M. & Wenz, H. Classification of sleep stages using multi-wavelet time frequency entropy and LDA. (2010).

20 Anderer, P. *et al.* Computer-assisted sleep classification according to the standard of the American Academy of Sleep Medicine: validation study of the AASM version of the Somnolyzer 24× 7. *Neuropsychobiology* **62**, 250-264 (2010).

21 Berthomier, C. *et al.* Automatic analysis of single-channel sleep EEG: validation in healthy individuals. *Sleep* **30**, 1587-1595 (2007).

22 Schaltenbrand, N. *et al.* Sleep stage scoring using the neural network model: comparison between visual and automatic analysis in normal subjects and patients. *Sleep* **19**, 26-35 (1996).

23 Biswal, S. *et al.* SLEEPNET: automated sleep staging system via deep learning. *arXiv preprint arXiv:1707.08262* (2017).

24 Al-Hussaini, I., Xiao, C., Westover, M. B. & Sun, J. SLEEPER: interpretable Sleep staging via Prototypes from Expert Rules. *arXiv preprint arXiv:1910.06100* (2019).

25 Schroff, F., Kalenichenko, D. & Philbin, J. in *Proceedings of the IEEE conference on computer vision and pattern recognition.* 815-823.

26 Jing, L. & Tian, Y. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

27 Wang, J., Jiao, J. & Liu, Y.-H. in *European Conference on Computer Vision.* 504-521 (Springer).

28 Fang, H. & Xie, P. CERT: Contrastive Self-supervised Learning for Language Understanding. *arXiv preprint arXiv:2005.12766* (2020).

29 Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. in *Advances in neural information processing systems.* 3111-3119.

30 Shukla, A., Petridis, S. & Pantic, M. Does Visual Self-Supervision Improve Learning of Speech Representations? *arXiv preprint arXiv:2005.01400* (2020).

31 Franceschi, J.-Y., Dieuleveut, A. & Jaggi, M. in *Advances in Neural Information Processing Systems.* 4650-4661.

32 Oord, A. v. d., Li, Y. & Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

33 Lemkhenter, A. & Favaro, P. Boosting Generalization in Bio-Signal Classification by Learning the Phase-Amplitude Coupling. *arXiv preprint arXiv:2009.07664* (2020).

34 Cheng, J. Y., Goh, H., Dogrusoz, K., Tuzel, O. & Azemi, E. Subject-aware contrastive learning for biosignals. *arXiv preprint arXiv:2007.04871* (2020).

35 Berry RB, B. R., Gamaldo C, Harding SM, Lloyd RM, Quan SF, Troester MT, Vaughn BV. AASM scoring manual updates for 2017 (version 2.4). *Journal of Clinical Sleep Medicine* (2017).

36 Gutmann, M. & Hyvärinen, A. in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics.* 297-304.

37 Zhang, G.-Q. *et al.* The National Sleep Research Resource: towards a sleep data commons. *Journal of the American Medical Informatics Association* **25**, 1351-1358 (2018).

38 Quan, S. F. *et al.* The sleep heart health study: design, rationale, and methods. *Sleep* **20**, 1077-1085 (1997).

39 Kemp, B., Zwinderman, A. H., Tuk, B., Kamphuisen, H. A. & Oberye, J. J. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG. *IEEE Transactions on Biomedical Engineering* **47**, 1185-1194 (2000).

40 Liu, X. *et al.* Self-supervised learning: Generative or contrastive. *arXiv preprint arXiv:2006.08218* **1** (2020).

41 McInnes, L., Healy, J. & Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).

42 Ioffe, S. & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

43 Clevert, D.-A., Unterthiner, T. & Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289* (2015).

44 He, K., Zhang, X., Ren, S. & Sun, J. in *Proceedings of the IEEE conference on computer vision and pattern recognition.* 770-778.

45 Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

IMAGE AND LEGENDS

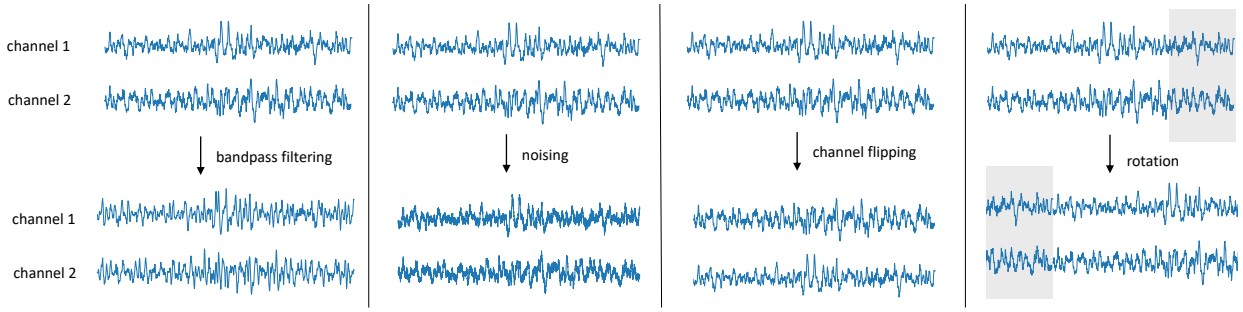

Figure 1. Data Augmentations for EEG Signals. The figure shows four different augmentation methods that we used for the study: bandpass filtering, signal noising, channel flipping and rotation.

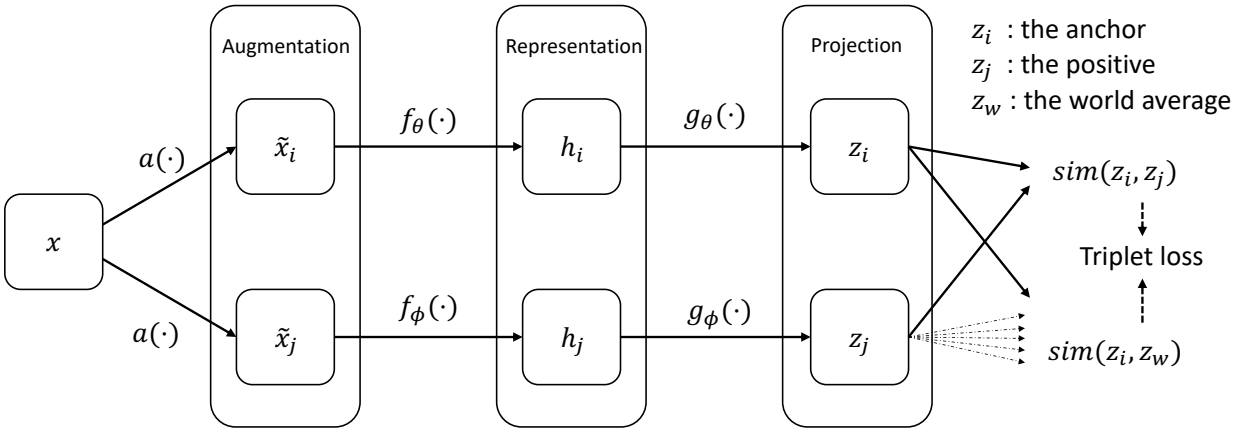

Figure 2. ContraWR Model Pipeline. We show the two-way model pipeline in this figure. The online network (upper) is updated by gradient descent, while the momentum target network (lower) is updated by exponential moving average (EMA). Finally, the results from two models form the contrastive loss function.

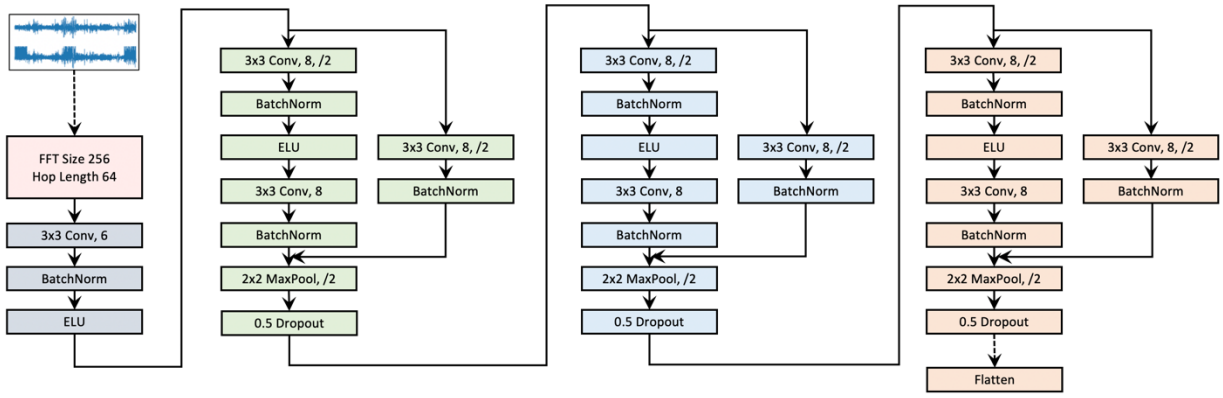

Figure 3. STFT Convolutional Encoder Network. The encoder network first transforms raw signals into spectrogram by STFT, then a CNN-based encoder is built on top of the spectrogram.

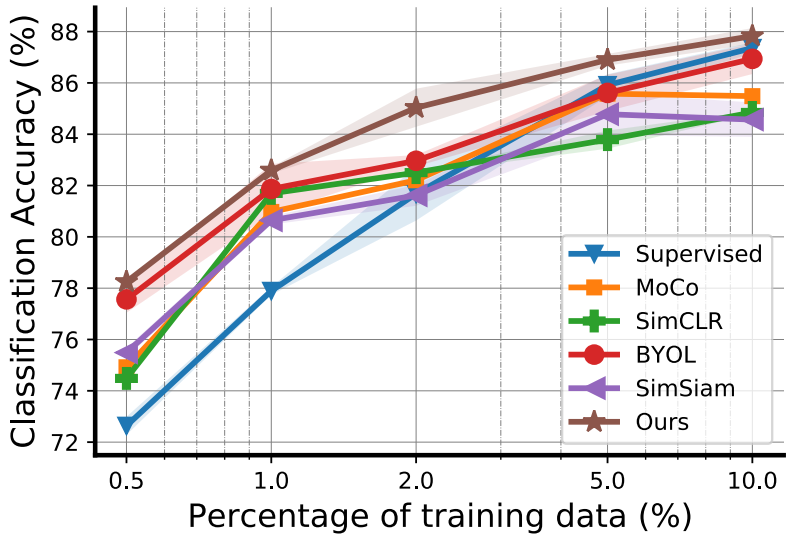

Figure 4. Model Performance with Different Amount of Training Data (on Sleep EDF). All models have the same encoder architecture. For the contrastive method, we train a logistic regression model on top of the frozen encoder, and for the supervised model, we train the encoder along with the final nonlinear classification layer from scratch. The amount of training data is set 0.5%, 1%, 2%, 5%, 10% for this experiment. Each configuration runs with 5 different random seeds and the error bars indicate the standard deviation over 5 experiments.

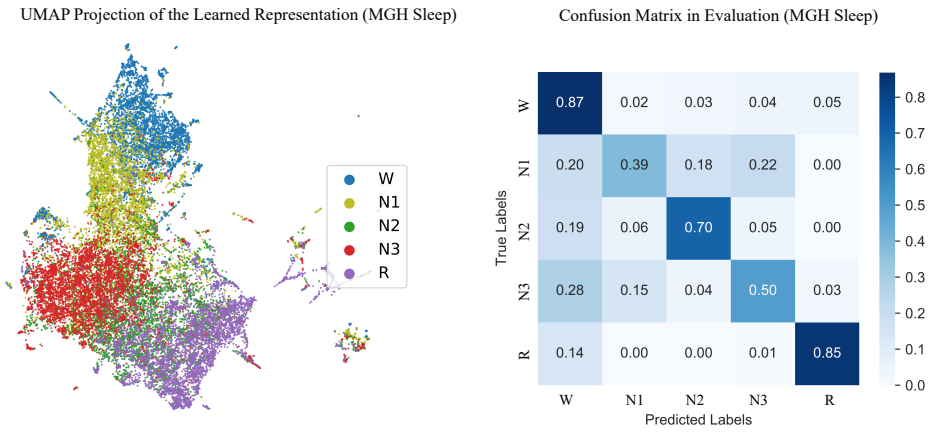

Figure 5. UMAP Projection and Confusion Matrix. Using MGH dataset, we project the output representations of each signal into indicative 2D visualization (left). We also show the confusion matrix of downstream classification task (right).

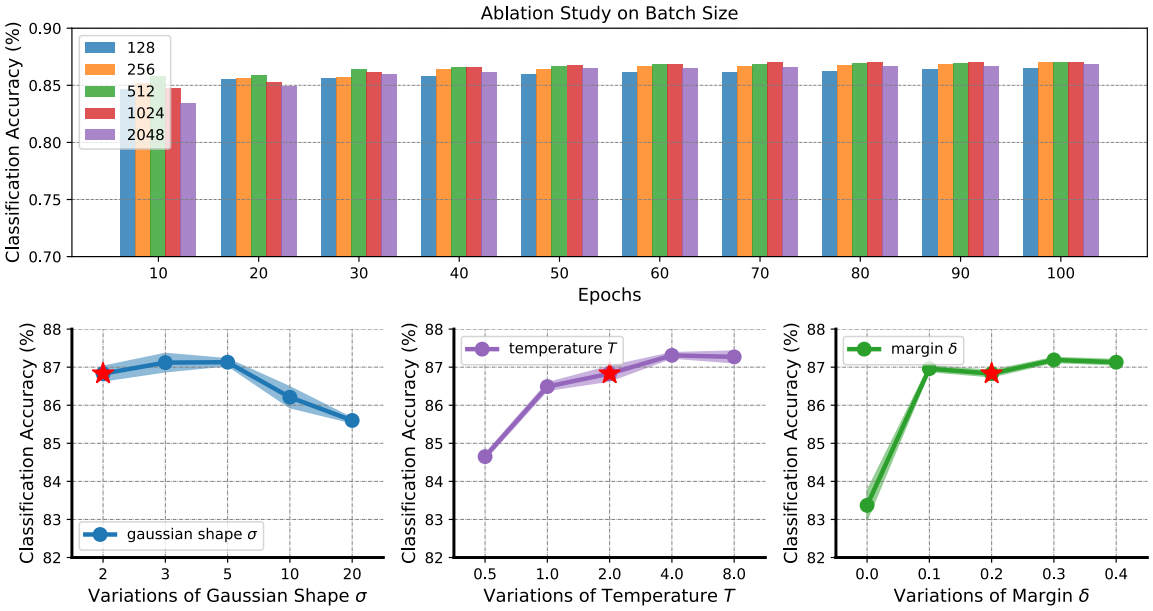

Figure 6. Ablation Study on Batch Size and Three Hyperparameters. The red star denotes the default setting. It is obvious that with a larger batch size, the model will perform better, while it is not sensitive to all hyperparameters. Each configuration runs with 5 different random seeds and the error bars indicate the standard deviation over 5 experiments.

TABLES

Table 1. Dataset Statistics

Name	Location	#channels	Frequency	#recordings	#epochs	Size
SHHS	at-home	2	125.0 Hz	5,445	4,535,949	260 GB
Sleep EDF	at-home	2	100.0 Hz	153	415,089	20 GB
MGH Sleep	in-lab	6	200.0 Hz	6,478	4,863,523	1,322 GB

Table 2. Class Label Distribution

Name	Awake (W)	Non-REM (N1)	Non-REM (N2)	Non-REM (N3)	REM (R)
SHHS	1,306,742 (28.8%)	169,021 (3.7%)	1,856,130 (40.9%)	571,191 (12.6%)	632,865 (14.0%)
Sleep EDF	285,561 (68.8%)	21,522 (5.2%)	69,132 (16.6%)	13,039 (3.2%)	25,835 (6.2%)
MGH Sleep	2,154,540 (44.3%)	481,488 (9.9%)	700,347 (14.4%)	855,980 (17.6%)	671,168 (13.8%)

Table 3. Sleep Staging Accuracy Comparison with Difference Methods (%)

Name	Sleep EDF	SHHS	MGH Sleep
Supervised	84.98 ± 0.3562	75.61 ± 0.9347	69.73 ± 0.4324
Untrained Encoder	77.83 ± 0.0232	60.03 ± 0.0448	55.64 ± 0.0082
MoCo	85.58 ± 0.7707	77.10 ± 0.2743	62.14 ± 0.7099
SimCLR	83.79 ± 0.3532	76.61 ± 0.3007	67.32 ± 0.7749
BYOL	85.61 ± 0.7080	76.64 ± 0.3783	70.75 ± 0.1461
SimSiam	84.78 ± 0.8028	74.25 ± 0.4796	62.08 ± 0.4902
ContraWR	85.94 ± 0.2326	77.52 ± 0.5748	71.97 ± 0.1774
ContraWR+	**86.90 ± 0.2288**	**77.97 ± 0.2693**	**72.03 ± 0.1823**

Table 4. Evaluation Accuracy of Different Augmentations (%)

Augmentations	Accuracy	Augmentations	Accuracy
bandpass	84.23 ± 0.2431	Bandpass + noising	85.37 ± 0.1214
noising	83.60 ± 0.1182	Noising + rotation	84.78 ± 0.1932
rotation	84.65 ± 0.2844	Rotation + bandpass	85.25 ± 0.1479
Bandpass + flipping	85.77 ± 0.2337	Bandpass + noising + flipping	85.76 ± 0.1794
Noising + flipping	84.45 ± 0.1420	Noising + rotation + flipping	85.17 ± 0.2301
Rotation + flipping	85.13 ± 0.0558	Rotation + bandpass + flipping	86.38 ± 0.2789

APPENDIX

Common Contrastive Methods and Limitations

Recent self-supervised methods use similar concepts as we described in the "Background and Concepts" section, and they can be categorized into two classes based on the framework. The first-class model adopts the two-encoder pipeline (such as our model, MoCo, BYOL), where the online encoder network is updated by gradient descent and the momentum encoder network is updated by exponential moving average (EMA). The second-class model uses one-encoder pipeline (such as SimCLR, SimSiam), where the same encoder network is used to encode two augmented versions, and it is updated by gradient descent. Recent self-supervised models have devised many different contrastive losses based on the anchor $\boldsymbol{z_i}$, the positive sample $\boldsymbol{z_j}$, and a set of random samples $\{\boldsymbol{z_k}\}$, where the random samples are usually conceived as negative samples. In the following, we present the most popular noise contrast estimation loss (NCE)[36].

In NCE loss, the distance between any two projections $(\boldsymbol{z_i}, \boldsymbol{z_j})$ are reflected by their measure of cosine similarity (since $\boldsymbol{z} \in \mathbb{S}^{m-1}, ||\boldsymbol{z}|| = 1$),

$$cos(\boldsymbol{z_i}, \boldsymbol{z_j}) = \langle \boldsymbol{z_i}, \boldsymbol{z_j} \rangle = 1 - \frac{1}{2}||\boldsymbol{z_i} - \boldsymbol{z_j}||^2.$$

Based on this, the NCE loss [3,7,16,32] is formulated: for one positive pair $(\boldsymbol{z_i}, \boldsymbol{z_j})$ and K "negative" pairs $(\boldsymbol{z_i}, \boldsymbol{z_k}), k = 1..K$, NCE loss is defined as,

$$\boldsymbol{L}(i,j) = -log\frac{exp(cos(\boldsymbol{z_i}, \boldsymbol{z_j}))}{exp(cos(\boldsymbol{z_i}, \boldsymbol{z_j})) + \sum_k exp(cos(\boldsymbol{z_i}, \boldsymbol{z_k}))}$$

By minimizing this loss, the distance between positive pairs $(\boldsymbol{z_i}, \boldsymbol{z_j})$ is minimized, while the anchor $\boldsymbol{z_i}$ would be far away from any "negative" projections $\boldsymbol{z_k}$ as possible from the dataset. However, the "negative" projections are randomly picked from the dataset, and they are conceived negative, which might not coincide with real labeling semantics. In fact, a "negative" sample $\boldsymbol{z_k}$ might be from the same class and close to the anchor $\boldsymbol{z_i}$ as well, especially in a noisy environment. This phenomenon is analyzed as a limitation of negative sampling[4,6], which hurts the performance significantly[5]. This paper is proposed to address this limitation.

Implementation

Data augmentation $a(\cdot)$. The human brain EEG signals are typically identified by different frequencies based on various sleep stages. In our paper, we consider the following four augmentation methods: **(i) Bandpass Filtering.** To reduce noise, we use the order-1 Butterworth filter (implemented by $scipy.signal.butter$), only the within-band frequency is preserved after augmentation. For Sleep EDF, we use frequency band interval (1, 5) and (30, 49); For SHHS, we use interval (1, 3) and (30, 60); For MGH Sleep, we only use interval (0, 30). **(ii) Noising.** We add independent and identically distributed high-frequency or low-frequency noise onto each channel. For three datasets, the high-frequency signal is sampled from uniform distribution modulated by a noise degree, following the equation:

$$noise\_seq = D * A * uniform\_random\_seq,$$

Where D is the noise degree (we use D=0.05 for all datasets), A is the amplitude range of the original signal, $uniform\_random\_seq$ is an independent and identically distributed (i.i.d.) sequence that has the same length as the signal and is generated from a uniform distribution in (-1,1) by $np.random.rand$. To generate the low-

frequency noise, we first sample a random noise sequence with $\frac{1}{100}$ of the signal length in the same way, and we later use $scipy.interpolate.interp1d$ to interpolate the noise sequence into the same length, whose frequency will turn low. After generating the noise sequence, we will add the noise to the original signal, where the probability of adding "high" or "low" or "both" will be equal. **(iii) Channel Flipping.** The sensor on the left side and the right of the brain are placed symmetrically. Thus, we flip the corresponding channels as another augmentation method. For Sleep EDF, we can flip the Fpz-Cz and Pz-Oz channels; For MGH Sleep, we can flip the F3-M2 and F4-M1, or C3-M2 and C4-M1 or O1-M2 and O2-M1; For SHHS, we can flip C3/A2 and C4/A1. **(iv) Rotation.** Within one instance, we will rotate/delay the signal for a certain time span. For three datasets, we uniformly split a signal epoch into two pieces and then resemble it as the augmentation. An illustration is provided in Figure 1. After doing augmentations, we will clip the signal amplitudes within a valid sensing range (which is the max measured signal amplitude, 2.5e-4 for Sleep EDF, 50 for MGH, 1.25e-4 for SHHS). To get an augmented version of a signal, we randomly apply one of the augmentation methods with equal probability.

In this paper, we extract information mainly from the frequency domain. For any sample $\boldsymbol{x}$, i.e., raw 30-second EEG signals, we take Short-Time Fourier Transforms (STFT) on each EEG channel. Importantly, when applying STFT transformation for each channel, we could extract both the amplitude and the phase information (they are both essential[33]), and then stack them together as two feature channels.

Network Architecture $f(\cdot)$. The implemented contrastive encoder is shown in Figure 3. Different datasets share a similar architecture with slightly different configurations since the signal input has different dimensions. The raw/augmented EEG signal instance first goes through an STFT module with 256 size and 64 hop length. The resulting STFT spectrogram passes through a convolutional layer with batch normalization[42] and ELU activation[43] and three residual blocks[44]. The latent representation is given by flattening the output of the last residual block.

Projector $g(\cdot)$**.** A recent work, SimCLR[3] shows that introducing a learnable nonlinear transformation between the latent representation and the contrastive loss will substantially improve the quality of the learned representations. In this work, the transformation, i.e., the projector, is implemented by a standard 2-layer fully connected network with ReLU as the activation. We further normalize the output by L2 norm.

All models are implemented using PyTorch 1.4.0 and optimized with Adam optimizer[45], 2e-4 as learning rate, 1e-4 as weight decay. For all datasets, we use 256 as batch size and use $\sigma = 2, \delta = 0.2, T = 2$ as the hyperparameters. In terms of the dimension of learned representations, 128 is for Sleep EDF, 256 for SHHS, and 192 for MGH. We implement the logistic regression model by scikit-learn with default setting and 500 as the maximum iteration for logistic regression evaluation of contrastive methods (the logistic regression is implemented by $klearn.linear\_model.LogisticRegression$ with $max\_iter = 500$ argument.). The experiments are conducted on two NVIDIA GTX 3090 GPUs with 24GB memory each, a 32-core CPU Linux machine with 256GB RAM. Note that the training process is IO-intensive, which involves loading sample files from the disk batch-by-batch. Therefore, a 4TB SSD persistent disk is used to store the raw signal epochs.