Proceedings of Machine Learning Research 136:238–253, 2020 Machine Learning for Health (ML4H) 2020

# Contrastive Representation Learning for Electroencephalogram Classification

**Mostafa 'Neo' Mohsenvand** MMV@MIT.EDU
*MIT Media Lab*

**Mohammad Rasool Izadi** MIZADI@ND.EDU
*University of Notre Dame*

**Pattie Maes** PATTIE@MEDIA.MIT.EDU
*MIT Media Lab*

**Editors:** Emily Alsentzer$^{\otimes}$, Matthew B. A. McDermott$^{\otimes}$, Fabian Falck, Suproteem K. Sarkar, Subhrajit Roy$^{\ddagger}$, Stephanie L. Hyland$^{\ddagger}$

## Abstract

Interpreting and labeling human electroencephalogram (EEG) is a challenging task requiring years of medical training. We present a framework for learning representations from EEG signals via contrastive learning. By recombining channels from multi-channel recordings, we increase the number of samples quadratically per recording. We train a channel-wise feature extractor by extending the SimCLR framework to time-series data. We introduce a set of augmentations for EEG and study their efficacy on different classification tasks. We demonstrate that the learned features improve EEG classification and significantly reduce the amount of labeled data needed on three separate tasks: (1) Emotion Recognition (SEED), (2) Normal/Abnormal EEG classification (TUH), and (3) Sleep-stage scoring (SleepEDF). Our models show improved performance over previously reported supervised models on SEED and SleepEDF and self-supervised models on all three tasks.

**Keywords:** EEG, Self-supervised Learning, Contrastive Learning, Emotion Recognition, Sleep-stage scoring, Abnormal EEG detection

## 1. Introduction

Electroencephalography (EEG) is a non-invasive technique for measuring the electrical activity of the brain. Since its invention in 1924, EEG has found many applications in clinical and research settings. EEG is obtained by placing an array of sensors on the skull and recording the voltage differences between the sensors. Despite the relative ease of recording, EEG signals are noisy, hard to interpret, and challenging to use in automated scenarios (e.g. via machine learning).

Traditionally, most machine learning applications on EEG signals have used handcrafted features inspired by the underlying neuroscientific findings. In recent years, representation learning and deep learning, in particular, have been applied to EEG signals and have lead to significant progress in many classification tasks (see Craik et al. (2019); Roy et al. (2019b)). Despite the success of deep learning in classifying EEG signals, the majority of these approaches learn in a supervised manner that restricts the use of the learned features to the specific task at hand. Moreover, labeling EEG data is cumbersome and requires either many years of medical training or sophisticated experimental design. Due to these problems, the amount of publicly available and labeled EEG data is limited and existing datasets are relatively small. Moreover, the existing datasets use incompatible EEG setups (e.g. different number of channels, sampling rates, types of sensors, etc.) that make them hard to fuse to obtain a larger dataset appropriate for unsupervised learning.

We are presenting a new framework that allows us to (1) combine multiple EEG datasets, (2) use the underlying physics of EEG signals to multiply the number of samples (quadratic increase), and (3) learn representations in a self-supervised manner via contrastive learning without requiring labels. Our approach concerns extracting features from a single channel at a time as opposed to considering all channels simultaneously. That allows us to recombine channels of a multi-channel recording to obtain new channels (see Section 2.1) and fuse multiple datasets with minimal preprocessing. In Section 3.5, we study the effect of channel recombination (CR) and dataset fusion (DF) through an ablation study. Our results show that CR and DF steps can significantly improve the quality of downstream tasks.

To learn EEG representations, we modify the SimCLR framework (Chen et al. (2020)) to work with time-series data. SimCLR is a contrastive learning method that learns representations that are invariant under a set of *augmentations* through a contrastive loss (see Section 2.3 and Figure 1.A). Recently, contrastive learning has been successful in computer vision for representation learning and pre-training (Bachman et al. (2019); He et al. (2019); Chen et al. (2020); Khosla et al. (2020)). In contrast to images where the set of augmentations are intuitive and easily verifiable by the human eye, it is not clear what augmentations could be beneficial for EEG. We consulted practicing neurologists and EEG researchers to select a set of transformations that leave the semantic information in EEG channels intact. We performed experiments and studied the efficacy of different augmentations in downstream tasks (See Sections 2.2 and 3.6).

Since different EEG classification tasks require signals of different lengths, we designed our method to output sequential representations (as opposed to point-representations) of equal length to the input signal (see Section 2.3).

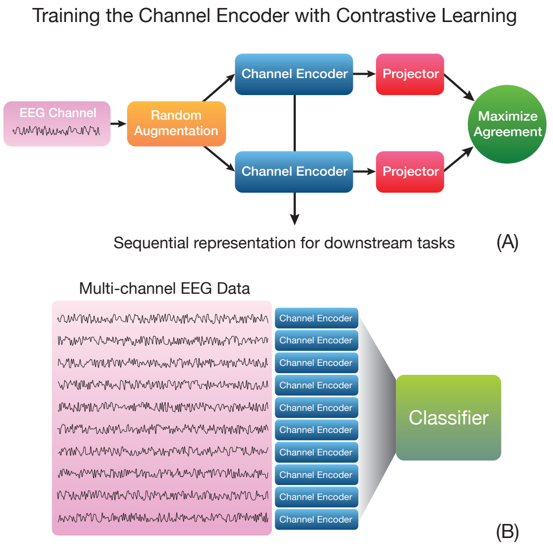


Figure 1: A. Pretraining the channel encoder. B. Using the pre-trained channel encoder to classify multi-channel EEG data

We applied our pre-trained channel encoder on three separate classification tasks: (1) Emotion Recognition on the SEED dataset (ER), (2) Normal/Abnormal Classification on the TUH dataset (NAC), and, (3) Sleep-stage scoring on the SleepEDF dataset (SSS). These tasks reflect a wide range of applications since each concerns an entirely different area of brain research. Our method showed improved performance over multiple previously reported supervised and self-supervised models, with remarkable sample-efficiency and achieved 85.77% on ER and 85.12% on SSS tasks when fine-tuned on all of the labels (see Section 3). In particular, we compared our results with 3 temporal contrastive models namely Contrastive Predictive Coding (CPC) (Oord et al. (2018); Banville et al. (2020)), Temporal Shuffling (TS) (Banville et al. (2020)) and Relative Positioning (RP) (Banville et al. (2020)) (see Section 4). Our models achieved significantly higher accuracy and sample-efficiency compared to CPC, TS, and RP on all three tasks.

At the end (Section 3.7), we study the structure of representations learned by our method, showing that the latent space is reasonably divided into physiologically meaningful regions.

## 2. Method

### 2.1. Channel recombination and preprocessing

Self-supervised learning via deep neural networks requires large amounts of data. Most EEG datasets are relatively small and incompatible with one another. However, if the goal is to learn the representation of a single channel, we can combine different datasets to obtain a larger one. Moreover, we can recombine channels in a multi-channel recording to obtain more valid channels. An EEG channel represents the voltage difference between a sensor and a common reference. By subtracting two channels, one obtains a new channel that represents the voltage difference between the two sensors, resulting in another physiologically valid channel. This is referred to as re-referencing and is frequently used by neurologists to obtain different views of data (Marcuse et al. (2015)). Figure 2 shows the process of recombining channels from a 3-channel recording. By including an extra common average channel and performing recombination, we obtain $n \times (n-1) + n = n^2$ new channels for an $n$-channel recording.

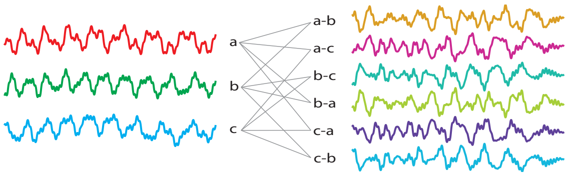


Figure 2: channel recombination for a 3-channel recording

**Datasets** Focusing on single channels enables us to fuse multiple datasets. By resampling the data to a fixed sampling rate (200 Hz), we combined multiple datasets: (1) Temple University Hospital Abnormal EEG Corpus (López et al. (2017)), (2) SEED Emotion Recognition dataset (Duan et al. (2013); Zheng and Lu (2015)), (3) Sleep EDF dataset (Kemp et al. (2000); Goldberger et al. (2000)), (4) Texas State University Resting State dataset (Trujillo et al. (2017)) and (5) ISRUC-Sleep dataset (Khalighi et al. (2016)). We selected datasets with raw or minimally processed signals that entailed long (¿20 seconds) sequences. In Section 3.5 we run an ablation study to understand the effect of recombining channels and fusing extra datasets on the final accuracy of downstream tasks.

**Preprocessing** We have resampled all datasets to 200Hz and applied a fifth-order band-pass Butterworth filter (0.3-80 Hz). We have also removed the channels that involved voltages higher than 500 $\mu$Vs as they normally represent artifacts. To train the encoder, we cut the channels into chunks of 20 seconds (see Section 3.1 for more details on the sequence length).

## 2.2. Channel augmentations

A key ingredient of contrastive learning is a set of augmentations (or transformations) that do not alter the semantic information of data. A contrastive learning algorithm learns representations that are maximally similar for augmented instances of the same data-point and minimally similar for different data-points. For example, rotating images or increasing the amplitude of audio signals can dramatically change the numerical values but not affect the meaning of data. By identifying a set of such augmentations, we can construct a self-supervised pretext task. Our objective is to learn features that are not sensitive to the transformations and reflect the high-level content of EEG signals.

We consulted four neurologists and two postdoctoral researchers at [anonymized hospital research group] specializing in clinical interpretation of EEG to identify a set of augmentations that do not change the interpretation of EEG data. From the list, we chose the transformations that were easy to randomize programmatically and ran preliminary experiments (see Appendix E) to choose a minimal effective set. Figure 3 shows an example of each of the selected transformations.

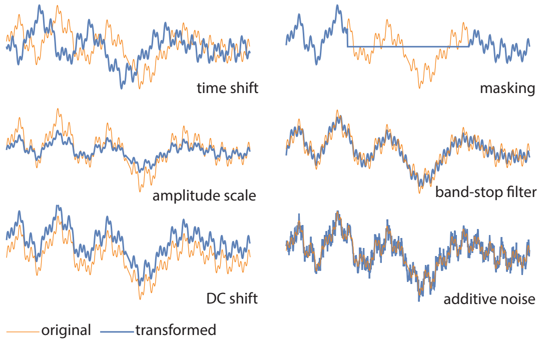


Figure 3: Channel Augmentations

The strength of each transformation is randomized on a specific range recommended by neurologists. Table 1 shows the range of transformations used to train the contrastive encoder.

Table 1: Transformation Ranges

| Transformation | min | max |
|---|---|---|
| amplitude scale | 0.5 | 2 |
| time shift (samples) | -50 | 50 |
| DC shift ($\mu$V) | -10 | 10 |
| zero-masking (samples) | 0 | 150 |
| additive Gaussian noise ($\sigma$) | 0 | 0.2 |
| band-stop filter (5 Hz width) (Hz) | 2.8 | 82.5 |

## 2.3. Learning algorithm

Our method learns features by maximizing the similarity between differently augmented transformations of the same channel through a contrastive loss in the latent space (Figure 1.A). We refer to our framework as SeqCLR (Sequential Contrastive Learning of Representations). Similar to SimCLR, our method contains four modules.

**Channel Augmenter** randomly transforms a mini-batch of $N$ channels into $2N$ augmented channels. For each channel, the module randomly applies two of the augmentations mentioned in Section 2.2 resulting in a positive pair.

**Channel Encoder** transforms an input channel into four feature channels of *the same length* (see Appendix Appendix B for hyper-parameter selection). This property enables us to encode sequences of different lengths for different downstream tasks. For instance, emotion recognition task is defined on 1-second-long segments, while the sleep staging task considers 30-second-long epochs. We designed two encoder architectures: (1) A recurrent encoder (Figure 4.A) with a multi-scale input (using downsampling and upsampling of the channel) to allow the GRU units to learn features at different time-scales. This architecture uses two recurrent residual units (2) A convolutional encoder (Figure 4.B) which utilizes reflection paddings (corresponding to the kernel size of the subsequent convolutional layer) to ensure the output signal is of the same length as the input signal. This architecture uses four convolutional residual units.

**Projector** A recurrent projection head that collapses the output of the encoder into a 32-dimensional point (Figure 4.C). The projector network uses downsampling and bi-directional LSTM units where the final outputs of each direction are concatenated and fed into dense layers with a ReLU activation in between.

**Contrastive Loss** identical to the NT-Xent (normalized temperature-scaled cross entropy) loss used in Chen et al. (2020). Given a set $\{\boldsymbol{x}_k\}$ including a positive pair of channels $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, the contrastive task aims to identify $\boldsymbol{x}_j$ in $\{\boldsymbol{x}_k\}_{k\neq i}$ for a given $\boldsymbol{x}_i$. Assuming that $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ are the outputs of the projector for the positive pair of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, the NT-Xent loss term for the positive pair is defined as:

$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k\neq i}^{2N} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)} \ ,$$

where $\text{sim}(\boldsymbol{u}, \boldsymbol{v})$ is the cosine similarity of $\boldsymbol{u}$ and $\boldsymbol{v}$ and $\tau$ is the temperature parameter. The final loss is the average of $\ell_{i,j}$ for all positive pairs in both orders ($i, j$ and $j, i$).

**Classifier** For downstream classification tasks, we discard the projector and use a classifier almost identical to the projector with two differences: (1) the output dimension of the last dense layer is set to the number of classes, and (2) a LogSoftmax layer is added afterward. We use a negative-log-likelihood loss alongside the LogSoftmax layer to compute the cross-entropy loss.

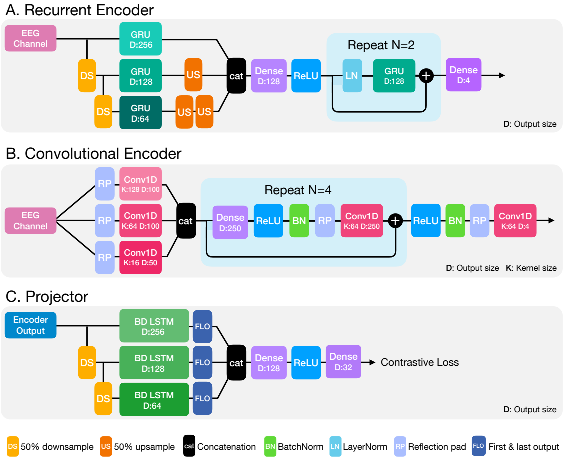


Figure 4: A. Recurrent encoder. B. Convolutional encoder, C. Projector

## 3. Experiments

In this section, we discuss the training process of the encoder and the results on three classification tasks. We also study sample-efficiency and the effect of augmentations on the quality of obtained features. For the classification tasks, we report the accuracy for inter-subject classification to compare the ability of our models to generalize across subjects. For more details about classification experiments, see Appendix D. We study the effect of dataset fusion and channel recombination on downstream tasks. We also study the structure of the latent space using dimensionality reduction.

### 3.1. Training the channel encoder

To train the encoder, we experimented with two architectures (Figure 4.A and 4.B). We chose the signal length of 20 seconds (4000 samples) for the channels by running preliminary classification experiments to decide the optimal signal length (see Appendix A). For longer sequences the contrastive loss falls rapidly during training since distinguishing between long samples is easier; however, the features learned on longer sequences, do not perform as well in classification. As discussed by Chen et al. (2020), this methodology benefits from larger batch sizes and training times. We used a batch size of 1000 for the convolutional encoder and 100 for the recurrent encoder. Both architectures were trained for 300 epochs. The $\tau$ hyperparameter was set to 0.05. We also used $\ell_2$ regularization with $\lambda = 10^{-4}$ (see Appendix C for hyperparameter selection).

### 3.2. Emotion recognition

**Dataset** We conducted experiments on SEED dataset (Duan et al. (2013); Zheng and Lu (2015)). The dataset involves EEG data of 15 subjects (7 males and 8 females) recorded in 62 channels. The data was recorded when participants watched emotional videos chosen from movies in three categories of emotions, namely *negative*, *neutral* and *positive*. The prior work on this dataset (Zheng and Lu (2015); Song et al. (2018); Li et al. (2018a,b, 2019); Zhong et al. (2020)) uses the hand-crafted Differential Entropy feature (Duan et al. (2013)) obtained from 1-second long epochs of EEG data on five frequency bands. Here we use the same signal length, but instead, the preprocessed EEG data was fed to the encoder.

**Results** We pass each channel through the encoder and concatenate the 4-dimensional output sequences. The input of the classifier, therefore, is a 4 × 62-dimensional sequence of length 200. We followed the same train/test partitioning protocol as Zheng and Lu (2015), which is also used in all other prior works.

Table 2: Emotion recognition on SEED

| Model | Accuracy | | | |
|---|---|---|---|---|
| Percentage of labels | 1% | 10% | 50% | 100% |
| RGNN | - | - | - | 85.30 |
| BiHDM | - | - | - | 85.40 |
| CPC | 69.17 | 76.33 | 79.98 | 81.12 |
| RP | 67.76 | 74.29 | 77.95 | 80.39 |
| TS | 69.73 | 78.27 | 81.66 | 82.10 |
| SeqCLR - C | 77.09 | 81.01 | 83.73 | 84.11 |
| SeqCLR - R | 76.52 | 79.04 | 81.45 | 83.78 |
| fine-tuned SeqCLR - C | **79.04** | **83.12** | **85.21** | **85.77** |
| fine-tuned SeqCLR - R | 78.18 | 82.93 | 84.00 | 85.25 |

We compared our results with two supervised models, RGNN (Zhong et al. (2020)), and, BiHDM (Li et al. (2019)) and three self-supervised models, Contrastive Predictive Coding (CPC) (Oord et al. (2018); Banville et al. (2020)), Temporal Shuffling (TS) Banville et al. (2020) and Relative Positioning (RP) Banville et al. (2020).

Table 2 shows the results of the experiments. The rows marked with SeqCLR-C (convolutional) and SeqCLR-R (recurrent) show the results without fine-tuning where the encoder parameters were frozen during training.

Our method improves other self-supervised algorithms by a large gap. Moreover, when fine-tuned on the entire dataset, SeqCLR achieves 85.77% accuracy, slightly higher than the current state of the art supervised model (BiHDM).

### 3.3. Normal vs. Abnormal Classification

**Dataset** We conduct experiments on Temple University Hospital (TUH) EEG Abnormal Corpus (López et al. (2017)). The dataset was created by selecting a demographically balanced subset of the larger TUH EEG Corpus through a manual review that consisted of 1488 abnormal and 1529 normal EEG sessions. The abnormalities

Table 3: Abnormality detection on TUH

| Model | Accuracy | | | |
|---|---|---|---|---|
| Percentage of labels | 1% | 10% | 50% | 100% |
| AlexNet | - | - | - | 87.32 |
| Alhussein et al. | - | - | - | **87.68** |
| CPC | 72.04 | 75.62 | 81.24 | 83.51 |
| RP | 75.17 | 77.13 | 82.16 | 83.37 |
| TS | 77.84 | 79.50 | 82.46 | 84.99 |
| SeqCLR - C | 78.53 | 85.44 | 85.52 | 86.27 |
| SeqCLR - R | 77.31 | 84.05 | 84.71 | 86.09 |
| fine-tuned SeqCLR - C | **83.19** | **86.98** | **87.21** | 87.45 |
| fine-tuned SeqCLR - R | 82.60 | 85.29 | 86.18 | 86.99 |

cover a wide range of conditions, including epilepsy, stroke, trauma, and coma. The dataset includes recordings with different numbers of channels and sampling rates, so we selected the 21 common channels amongst all recordings and downsampled to 200 Hz. Most of The prior work considers the first minute of the recordings for classification as the quality of signals drops with time due to the drying of sensors and sweating. However, Roy et al. (2019a) demonstrated that using the first 11 minutes of recordings can improve the classification results. We have also used the first 11 minutes in 1-minute long chunks.

**Results** The input of the classifier is a 4 × 21-dimensional sequence of length 12000 samples. We followed the same evaluation protocol as López et al. (2017) that is also used in all other prior work.

Table 3 shows the classification accuracy of our model compared to two supervised models: a variation of AlexNet (Amin et al. (2019)), and Alhussein et al. (2019) and three self-supervised models: CPC, TS, and RP.

Our method improves other self-supervised algorithms by a consistent gap and achieves near SOTA accuracy when fine-tuned on all of the labels.

### 3.4. Sleep Stage Classification

**Dataset** The expanded SleepEDF dataset (Goldberger et al. (2000)) involves EEG recordings of 20 healthy subjects during sleep. Each recording includes two-channel EEG data from Fpz-Cz and Pz-Oz with a sampling rate of 100 Hz. Each 30-second epoch was labeled with one of five classes (*Wake, REM, N1, N2, N3*) standing for sleep stages. We upsampled the signals to 200 Hz and only used the Fpz-Cz channel for a fair comparison with prior work. Similar to Tsinalis et al. (2016) and Vilamala et al. (2017), we used five 30-second epochs (2 before and 2 after for the context). Other methods have used more (e.g. Back et al. (2019)) or less (e.g. Phan et al. (2018)) number of epochs.

**Results** The input of the classifier is a 4-dimensional sequence of length 30000 samples. We followed the same evaluation protocol as Tsinalis et al. (2016) which is also used similarly in all other prior work.

We compare our results to two supervised models: DeepSleepNet (Supratak et al. (2017)), and, IITNET (Back et al. (2019)). U-Time Perslev et al. (2019), on the other hand, reports a higher average F1-score than IITNET and DeepSleepNet but does not report accuracy. We also compare our results to CPC, TS, and RP. Table 4 shows the results.

Our method improves other self-supervised algorithms by a large gap. Moreover, when fine-tuned on the entire dataset, SeqCLR achieves 85.12% accuracy, higher than both supervised models.

### 3.5. Ablation study of channel recombination and dataset fusion

As discussed in Section 2.1, we used channel recombination (CR) and dataset fusion (DF) to obtain a larger training set for self-supervised learning. Table 5 shows the effect

Table 4: Sleep-staging on SleepEDF

| Model | Accuracy | | | |
|---|---|---|---|---|
| Percentage of labels | 1% | 10% | 50% | 100% |
| DeepSleepNet | - | - | - | 82.00 |
| IITNET | - | - | - | 84.00 |
| CPC | 66.79 | 75.62 | 76.55 | 79.38 |
| RP | 68.04 | 74.00 | 77.84 | 79.15 |
| TS | 67.32 | 74.36 | 76.16 | 79.93 |
| SeqCLR - C | 71.09 | 73.63 | 81.18 | 82.71 |
| SeqCLR - R | 72.40 | 76.81 | 82.91 | 83.05 |
| fine-tuned SeqCLR - C | 74.16 | 81.81 | 82.60 | 83.91 |
| fine-tuned SeqCLR - R | **74.33** | **82.03** | **83.72** | **85.12** |

of removing each of these steps in the accuracy of the classifiers without fine-tuning.

Table 5: Ablation study of CR and DF

| Channel recombination | Dataset fusion | SEED | TUH | SleepEDF |
|---|---|---|---|---|
| ✗ | ✗ | 78.93 | 79.12 | 77.72 |
| ✓ | ✗ | 83.01 | 83.78 | 81.10 |
| ✗ | ✓ | 80.23 | 83.44 | 79.59 |
| ✓ | ✓ | **84.11** | **86.27** | **83.05** |

We observe that both steps of DF and CR improve the accuracy of the classifiers. In particular removing channel recombination had a stronger effect in all three tasks.

### 3.6. Ablation study of augmentations

To study the effect of each augmentation on the quality of the features, we ran an ablation study. We trained 6 versions of the encoder, removing one augmentations at a time. We trained the classifiers on 100% of the labels. Figure 5 shows the results for our best performing models. We have used the convolutional architecture for Emotion recognition and Normal/Abnormal classification and the recurrent architecture for Sleep-stage scoring. We note that masking and scaling are the most effective augmentations across the three classification tasks. On the other hand, additive noise and DC shift have the least effect on the performance of the classifiers.

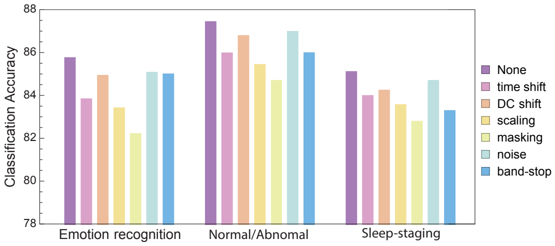


Figure 5: Ablation study of augmentations. Each bar shows the accuracy of the classifier when that augmentation is removed.

### 3.7. Study of the latent space

SeqCLR produces sequential representations that make the study of the latent space difficult. Instead of using the output sequences, we used the output of the projector. We fed the encoder a uniform sample of signals that were classified correctly (to avoid corrupted channels). To make the results comparable, we only chose the Fpz-Cz channel from each recording (total of 40,000 32-d points). Figure 6 shows the t-SNE representation (Maaten and Hinton (2008)) of the combined dataset obtained with the perplexity of 125. We observe that the channel encoder has reasonably partitioned the latent space into physiologically meaningful regions.

## 4. Related work

Self-supervised learning and unsupervised learning, in general, have had limited success with EEG signals. Denoising autoencoders have been used for pre-training and extracting features in studies such as Li et al. (2015) and Yin and Zhang (2017) and Yang et al. (2019). Restricted Boltzmann machines and

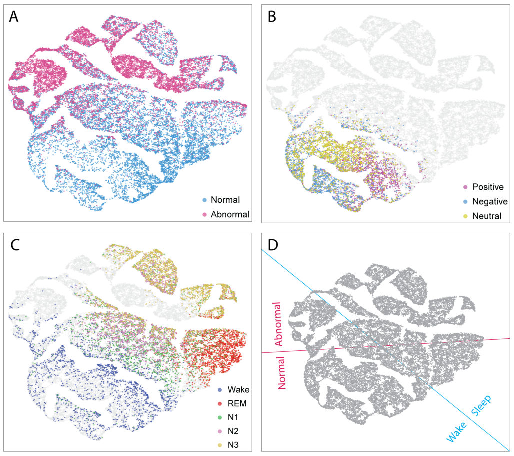


Figure 6: t-SNE plot for 40k Fpz-Cz channels sampled from A. THU, B. SEED and C. SleepEDF, D. Partitioning

deep belief networks (DBNs) have also been used for pre-training of supervised models. For instance Zheng et al. (2014) and Hassan et al. (2019) used DBN pre-training for emotion recognition.

Aside from EEG signals, contrastive learning has been used for other time-series data. Hyvarinen and Morioka (2016) introduced the *Time-contrastive learning* (TCL) that explores the temporal non-stationarity of time series data. The learned representations are optimized to discriminate data from different time segments. Specifically, the TCL network is trained to discriminate each segment of the time-series by using the segment IDs as labels. Hyvarinen and Morioka (2016) used TCL for classifying different brain states from magnetoencephalogram (MEG) signals.

Oord et al. (2018) proposed the *Contrastive Predictive Coding* (CPC) algorithm that learns representations by discriminating between possible future segments in time-series data for a given segment.

The most relevant work to ours is the study done by Banville et al. (2020). To learn representations for EEG signals, they made use of three self-supervised pretext tasks inspired by Hyvarinen and Morioka (2016) and Oord et al. (2018). CPC and two temporal context prediction tasks *Relative Positioning*(RP) and *Temporal Shuffling*(TS), were used for feature learning. They showed that the learned features are physiologically meaningful and used them for Sleep-stage scoring and Normal/Abnormal classification. Their experiments also revealed that self-supervised pretraining can greatly benefit sample-efficiency in EEG classification. Our paper also confirms their results and improves upon them. We show that our pretext task (SeqCLR), which is inspired by an image-based method (SimCLR), can outperform temporal pretext tasks and compete with the state-of-the-art supervised models.

## 5. Conclusion

We introduced SeqCLR, a self-supervised framework for learning representations of EEG signals. In doing so, we proposed a method to quadratically increase the number of samples per recording and fuse multiple datasets. We generalized the SimCLR framework for time-series data and used it to improve sample-efficiency and classification accuracy in 3 separate tasks of emotion recognition on the SEED dataset, normal/abnormal classification on the TUH dataset, and sleep-stage scoring on the SleepEDF dataset. Our models achieved improved performance over baseline self-supervised models. With fine-tuning on all of the labels, SeqCLR could achieve accuracies higher than the current state of the art supervised models in emotion recognition and sleep-staging tasks.

To train SecCLR, we introduced six augmentations on EEG channels. Our study shows two of these augmentations, namely, masking and scaling, play a crucial role in extracting useful features for downstream tasks.

Our results demonstrate that self-supervised learning techniques and contrastive learning, in particular, are promising tools for learning representations from EEG signals.

## References

Musaed Alhussein, Ghulam Muhammad, and M Shamim Hossain. Eeg pathology detection based on deep learning. *IEEE Access*, 7:27781–27788, 2019.

Syed Umar Amin, M Shamim Hossain, Ghulam Muhammad, Musaed Alhussein, and Md Abdur Rahman. Cognitive smart healthcare for pathology detection and monitoring. *IEEE Access*, 7:10745–10753, 2019.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15509–15519, 2019.

Seunghyeok Back, Seongju Lee, Hogeon Seo, Deokhwan Park, Tae Kim, and Kyoobin Lee. Intra-and inter-epoch temporal context network (iitnet) for automatic sleep stage scoring. *arXiv preprint arXiv:1902.06562*, 2019.

Hubert Banville, Omar Chehab, Aapo Hyvärinen, Denis-Alexander Engemann, and Alexandre Gramfort. Uncovering the structure of clinical eeg signals with self-supervised learning. *arXiv preprint arXiv:2007.16104*, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020.

Alexander Craik, Yongtian He, and Jose L Contreras-Vidal. Deep learning for electroencephalogram (eeg) classification tasks: a review. *Journal of neural engineering*, 16(3):031001, 2019.

Ruo-Nan Duan, Jia-Yi Zhu, and Bao-Liang Lu. Differential entropy feature for EEG-based emotion classification. In *6th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 81–84. IEEE, 2013.

Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

Mohammad Mehedi Hassan, Md Golam Rabiul Alam, Md Zia Uddin, Shamsul Huda, Ahmad Almogren, and Giancarlo Fortino. Human emotion recognition using deep belief network architecture. *Information Fusion*, 51:10–18, 2019.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

Aapo Hyvarinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. In *Advances in Neural Information Processing Systems*, pages 3765–3773, 2016.

Bob Kemp, Aeilko H Zwinderman, Bert Tuk, Hilbert AC Kamphuisen, and Josefien JL Oberye. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the eeg. *IEEE Transactions on Biomedical Engineering*, 47(9): 1185–1194, 2000.

Sirvan Khalighi, Teresa Sousa, José Moutinho Santos, and Urbano Nunes. Isruc-sleep: a comprehensive public dataset for sleep researchers. *Computer* *methods and programs in biomedicine*, 124:180–192, 2016.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

He Li, Yi-Ming Jin, Wei-Long Zheng, and Bao-Liang Lu. Cross-subject emotion recognition using deep adaptation networks. In *International Conference on Neural Information Processing*, pages 403–413. Springer, 2018a.

Junhua Li, Zbigniew Struzik, Liqing Zhang, and Andrzej Cichocki. Feature learning from incomplete eeg with denoising autoencoder. *Neurocomputing*, 165:23–31, 2015.

Yang Li, Wenming Zheng, Yuan Zong, Zhen Cui, Tong Zhang, and Xiaoyan Zhou. A bi-hemisphere domain adversarial neural network model for eeg emotion recognition. *IEEE Transactions on Affective Computing*, 2018b.

Yang Li, Wenming Zheng, Lei Wang, Yuan Zong, Lei Qi, Zhen Cui, Tong Zhang, and Tengfei Song. A novel bi-hemispheric discrepancy model for eeg emotion recognition. *arXiv preprint arXiv:1906.01704*, 2019.

Silvia López, I Obeid, and J Picone. Automated interpretation of abnormal adult electroencephalograms. *MS Thesis, Temple University*, 2017.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov): 2579–2605, 2008.

Lara V Marcuse, Madeline C Fields, and Jiyeoun Jenna Yoo. *Rowan's Primer of EEG E-Book*. Elsevier Health Sciences, 2015.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Mathias Perslev, Michael Jensen, Sune Darkner, Poul Jørgen Jennum, and Christian Igel. U-time: A fully convolutional network for time series segmentation applied to sleep staging. In *Advances in Neural Information Processing Systems*, pages 4415–4426, 2019.

Huy Phan, Fernando Andreotti, Navin Cooray, Oliver Y Chén, and Maarten De Vos. Joint classification and prediction cnn framework for automatic sleep stage classification. *IEEE Transactions on Biomedical Engineering*, 66(5):1285–1296, 2018.

Subhrajit Roy, Isabell Kiral-Kornek, and Stefan Harrer. Chrononet: a deep recurrent neural network for abnormal eeg identification. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 47–56. Springer, 2019a.

Yannick Roy, Hubert Banville, Isabela Albuquerque, Alexandre Gramfort, Tiago H Falk, and Jocelyn Faubert. Deep learning-based electroencephalography analysis: a systematic review. *Journal of neural engineering*, 16(5):051001, 2019b.

Tengfei Song, Wenming Zheng, Peng Song, and Zhen Cui. Eeg emotion recognition using dynamical graph convolutional neural networks. *IEEE Transactions on Affective Computing*, 2018.

Akara Supratak, Hao Dong, Chao Wu, and Yike Guo. Deepsleepnet: a model for automatic sleep stage scoring based on raw single-channel eeg. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(11):1998–2008, 2017.

Logan T Trujillo, Candice T Stanfield, and Ruben D Vela. The effect of electroencephalogram (eeg) reference choice on information-theoretic measures of the complexity and integration of eeg signals. *Frontiers in neuroscience*, 11:425, 2017.

Orestis Tsinalis, Paul M Matthews, and Yike Guo. Automatic sleep stage scoring using time-frequency analysis and stacked sparse autoencoders. *Annals of biomedical engineering*, 44(5):1587–1597, 2016.

Albert Vilamala, Kristoffer H Madsen, and Lars K Hansen. Deep convolutional neural networks for interpretable analysis of eeg sleep stage scoring. In *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2017.

Shuo Yang, Zhong Yin, Yagang Wang, Wei Zhang, Yongxiong Wang, and Jianhua Zhang. Assessing cognitive mental workload via eeg signals and an ensemble deep learning classifier based on denoising autoencoders. *Computers in biology and medicine*, 109:159–170, 2019.

Zhong Yin and Jianhua Zhang. Cross-session classification of mental workload levels using eeg and an adaptive deep learning model. *Biomedical Signal Processing and Control*, 33:30–47, 2017.

Wei-Long Zheng and Bao-Liang Lu. Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks. *IEEE Transactions on Autonomous Mental Development*, 7(3):162–175, 2015. doi: 10.1109/TAMD.2015.2431497.

Wei-Long Zheng, Jia-Yi Zhu, Yong Peng, and Bao-Liang Lu. Eeg-based emotion classification using deep belief networks. In *2014 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2014.

Peixiang Zhong, Di Wang, and Chunyan Miao. Eeg-based emotion recognition using regularized graph neural networks. *IEEE Transactions on Affective Computing*, 2020.

## Appendix A. Choosing the sequence-length

Different EEG classification tasks require different lengths of channels. For instance, classifying stages of sleep normally requires long sequences of 30-second or longer length. On the other hand, tasks such as emotion recognition or motor imagery classification are defined on shorter sequences of one-second or sometimes shorter. To ensure that the encoder learns futures that are useful for a wide range of tasks, we ran a preliminary experiment with two classification tasks: (1) emotion recognition and (2) sleep-stage scoring where we trained the encoder on signals of various length and compared the classification accuracy. Figure 7 shows the results for 8 different lengths.

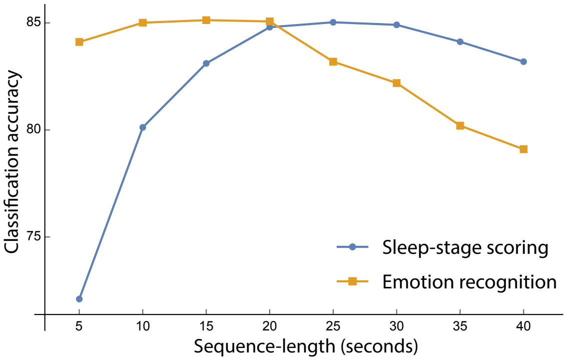


Figure 7: Classification accuracy when the encoder is trained on sequences of different length.

We observed that sequences of length 20-seconds perform well for both tasks. Increasing the length of the sequence improves the accuracy in emotion recognition slightly while decaying the performance in sleep-stage scoring.

## Appendix B. Choosing latent size

We trained the encoder with different latent dimensions (Figure 8). When channel encoder is used in fine-tuning for down-stream tasks, increasing the number of dimensions increases the number of parameters and the compute time linearly. Therefore we picked D=4 since it produces comparable accuracies while allowing the network to be fast in train-time and test-time.

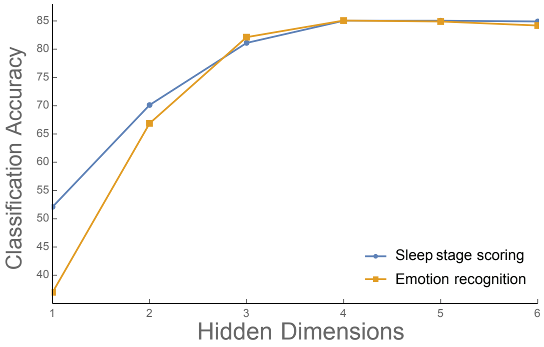


Figure 8: Classification accuracy when the encoder is trained with different latent dimensions

## Appendix C. Choosing $\tau$ and $\lambda$

We jointly optimized $\tau$ (temperature parameter of the contrastive loss) and $\lambda$ ($\ell_2$ regularization coefficient or weight-decay). We used the sleep-staging task on SleepEDF to select these parameters. The selected parameters worked comparably well in the other two tasks.

## Appendix D. Classification experiments

### D.1. Classifier architecture

The architecture of the classifier is very similar to the projector. The input is the con-

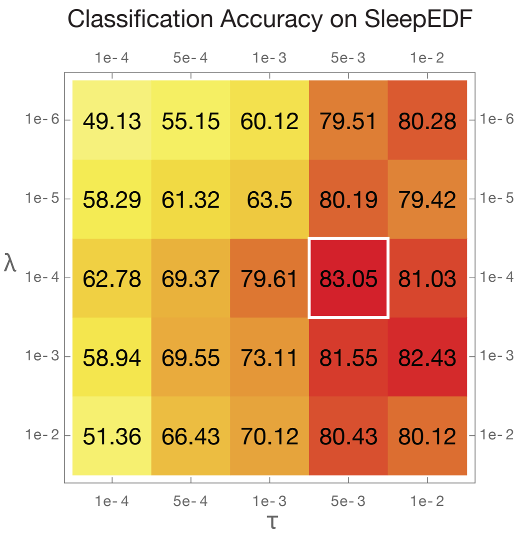


Figure 9: Classification accuracy when the encoder is trained with different values of $\tau$,\$$\lambda$

catenation of output of the encoder for all input channels of a multi-channel recording. Figure 10 shows the architecture.

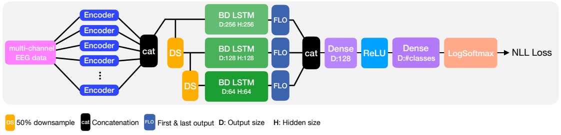


Figure 10: Classifier architecture. Note that the size of the output of the last dense layer is set to the number of classes.

### D.2. Training

All classifiers were trained with early stopping on the validation set. We performed the experiments on 8 Nvidia Titan-V GPUs. In half of the experiments, the parameters of the encoder were frozen. The training time is an order of magnitude longer when fine-tuning with the recurrent encoder. Table 6 shows the number of epochs and the batch-sizes used in training the classifiers.

Table 6: Training specifications

| Task | epochs | batch-size |
|---|---|---|
| Emotion recognition | 132 | 128 |
| Normal/Abnormal classification | 67 | 32 |
| Sleep-stage scoring | 29 | 64 |

### D.3. Confusion matrices

Here we report the confusion matrices for the three classification tasks with fine-tuning on 100% of the labels.

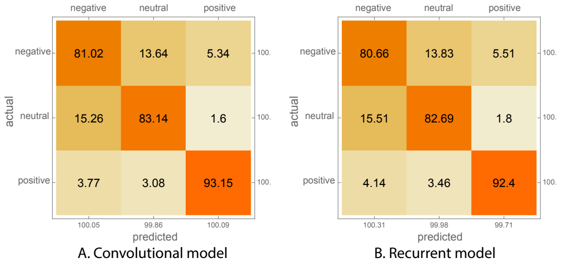


Figure 11: Confusion matrix for emotion recognition on SEED dataset with fine-tuning on 100% of the data. A. convolutional model, B. recurrent model

## Appendix E. Choosing effective augmentations

We set up a classification task with the convolutional SeqCLR architecture, only using a single augmentation at a time. We trained nine encoders and tested them on the three classification tasks mentioned in the main text. For training the classifiers, we froze

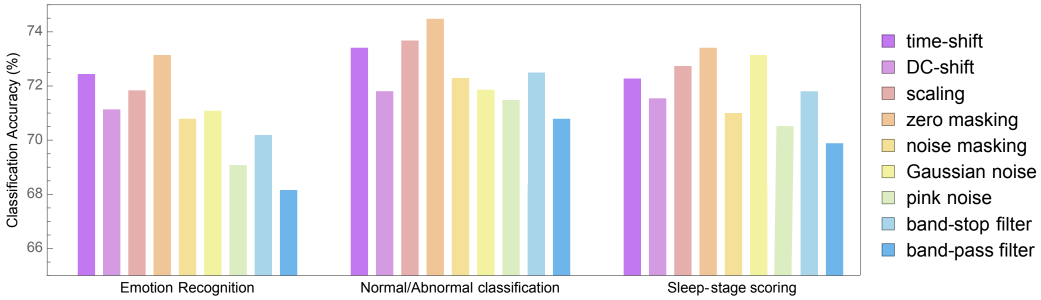
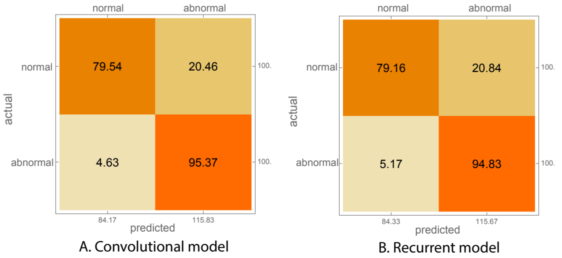


Figure 12: Classification accuracy of the models trained with one augmentation



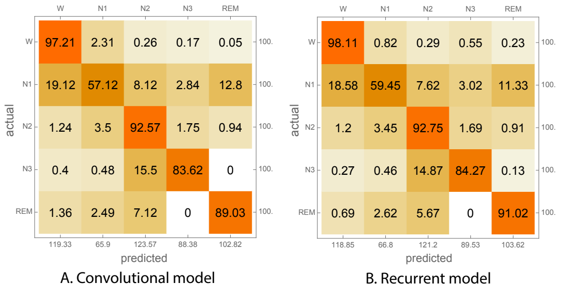
A. Convolutional model

| actual \ predicted | normal | abnormal | |
|---|---|---|---|
| normal | 79.54 | 20.46 | 100. |
| abnormal | 4.63 | 95.37 | 100. |
| | 84.17 | 115.83 | |

B. Recurrent model

| actual \ predicted | normal | abnormal | |
|---|---|---|---|
| normal | 79.16 | 20.84 | 100. |
| abnormal | 5.17 | 94.83 | 100. |
| | 84.33 | 115.67 | |

Figure 13: Confusion matrix for Normal/Abnormal classification on TUH dataset with fine-tuning on 100% of the data. A. convolutional model, B. recurrent model



A. Convolutional model

| actual \ predicted | W | N1 | N2 | N3 | REM | |
|---|---|---|---|---|---|---|
| W | 97.21 | 2.31 | 0.26 | 0.17 | 0.05 | 100. |
| N1 | 19.12 | 57.12 | 8.12 | 2.84 | 12.8 | 100. |
| N2 | 1.24 | 3.5 | 92.57 | 1.75 | 0.94 | 100. |
| N3 | 0.4 | 0.48 | 15.5 | 83.62 | 0 | 100. |
| REM | 1.36 | 2.49 | 7.12 | 0 | 89.03 | 100. |
| | 119.33 | 65.9 | 123.57 | 88.38 | 102.82 | |

B. Recurrent model

| actual \ predicted | W | N1 | N2 | N3 | REM | |
|---|---|---|---|---|---|---|
| W | 98.11 | 0.82 | 0.29 | 0.55 | 0.23 | 100. |
| N1 | 18.58 | 59.45 | 7.62 | 3.02 | 11.33 | 100. |
| N2 | 1.2 | 3.45 | 92.75 | 1.69 | 0.91 | 100. |
| N3 | 0.27 | 0.46 | 14.87 | 84.27 | 0.13 | 100. |
| REM | 0.69 | 2.62 | 5.67 | 0 | 91.02 | 100. |
| | 118.85 | 66.8 | 121.2 | 89.53 | 103.62 | |

Figure 14: Confusion matrix for Sleep-stage scoring on SleepEDF dataset with fine-tuning on 100% of the data. A. convolutional model, B. recurrent model

the encoder parameters. This experiment allowed us to choose the most effective augmentations. Figure 12 demonstrates the classification accuracy for each model.

We observed that the six augmentations, namely (1) zero-masking, (2) amplitude scaling, (3) time-shift, (4) Gaussian noise, (5) DC-shift, and (6) band-stop filter perform significantly better in extracting useful features for the downstream tasks.