Uncovering the structure of clinical EEG signals with self-supervised learning

Banville, Hubert

2021

Banville , H , Chehab , O , Hyvarinen , A , Engemann , D-A & Gramfort , A 2021 , ' Uncovering the structure of clinical EEG signals with self-supervised learning ' , Journal of Neural Engineering , vol. 18 , no. 4 , 046020 . https://doi.org/https://arxiv.org/abs/2007.16104 , https://doi.org/10.1088/1741-2552/abca18

http://hdl.handle.net/10138/342193
https://doi.org/https://arxiv.org/abs/2007.16104

Uncovering the structure of clinical EEG signals with self-supervised learning

Hubert Banville*[1,2], Omar Chehab[1], Aapo Hyvärinen[1,3], Denis-Alexander Engemann[1,4], and Alexandre Gramfort[1]

[1]Université Paris-Saclay, Inria, CEA, Palaiseau, France
[2]InteraXon Inc., Toronto, Canada
[3]Dept. of CS and HIIT, University of Helsinki, Finland
[4]Max Planck Institute for Human Cognitive and Brain Sciences, Department of Neurology, Leipzig, Germany

Abstract

Objective. Supervised learning paradigms are often limited by the amount of labeled data that is available. This phenomenon is particularly problematic in clinically-relevant data, such as electroencephalography (EEG), where labeling can be costly in terms of specialized expertise and human processing time. Consequently, deep learning architectures designed to learn on EEG data have yielded relatively shallow models and performances at best similar to those of traditional feature-based approaches. However, in most situations, unlabeled data is available in abundance. By extracting information from this unlabeled data, it might be possible to reach competitive performance with deep neural networks despite limited access to labels. *Approach.* We investigated self-supervised learning (SSL), a promising technique for discovering structure in unlabeled data, to learn representations of EEG signals. Specifically, we explored two tasks based on temporal context prediction as well as contrastive predictive coding on two clinically-relevant problems: EEG-based sleep staging and pathology detection. We conducted experiments on two large public datasets with thousands of recordings and performed baseline comparisons with purely supervised and hand-engineered approaches. *Main results.* Linear classifiers trained on SSL-learned features consistently outperformed purely supervised deep neural networks in low-labeled data regimes while reaching competitive performance when all labels were available. Additionally, the embeddings learned with each method revealed clear latent structures related to physiological and clinical phenomena, such as age effects. *Significance.* We demonstrate the benefit of self-supervised learning approaches on EEG data. Our results suggest that SSL may pave the way to a wider use of deep learning models on EEG data.

Keywords Self-supervised learning, representation learning, machine learning, electroencephalography, sleep staging, pathology detection, clinical neuroscience

1 Introduction

Electroencephalography (EEG) and other biosignal modalities have enabled numerous applications inside and outside of the clinical domain, e.g., studying sleep patterns and their disruption [1],

*hubert.jacob-banville@inria.fr

monitoring seizures [2] and brain-computer interfacing [3]. In the last few years, the availability and portability of these devices has increased dramatically, effectively democratizing their use and unlocking the potential for positive impact on people's lives [4, 5]. For instance, applications such as at-home sleep staging and apnea detection, pathological EEG detection, mental workload monitoring, etc. are now entirely possible.

In all these scenarios, monitoring modalities generates an ever-increasing amount of data which needs to be interpreted. Therefore, predictive models that can classify, detect and ultimately "understand" physiological data are required. Traditionally, this type of modelling has mostly relied on supervised approaches, where large datasets of annotated examples are required to train models with high performance.

However, obtaining accurate annotations on physiological data can prove expensive, time consuming or simply impossible. For example, annotating sleep recordings requires trained technicians to go through hours of data visually and label 30-s windows one-by-one [6]. Clinical recordings such as those used to diagnose epilepsy or brain lesions must be reviewed by neurologists, who might not always be available. More broadly, noise in the data and the complexity of brain processes of interest can make it difficult to interpret and annotate EEG signals, which can lead to high inter-rater variability, i.e., label noise [7, 8]. Furthermore, in some cases, knowing exactly what the participants were thinking or doing in cognitive neuroscience experiments can be challenging, making it hard to obtain accurate labels. In imagery tasks, for instance, the subjects might not be following instructions or the process under study might be difficult to quantify objectively (e.g., meditation, emotions). Therefore, a new paradigm that does not rely primarily on supervised learning is necessary for making use of large unlabeled sets of recordings such as those generated in the scenarios described above. However, traditional unsupervised learning approaches such as clustering and latent factor models do not offer fully satisfying answers as their performance is not as straightforward to quantify and interpret as supervised ones.

"Self-supervised learning" (SSL) is an unsupervised learning approach that learns representations from unlabeled data, exploiting the structure of the data to provide supervision [9]. By reframing an unsupervised learning problem as a supervised one, SSL allows the use of standard, better understood optimization procedures. SSL comprises a "pretext" and a "downstream" task. The downstream task is the task one is actually interested in but for which there are limited or no annotations. The pretext task, on the other hand, must be sufficiently related to the downstream task such that similar representations should be employed to carry it out; importantly, it must be possible to generate the annotations for this pretext task using the unlabeled data alone. For example, in a computer vision scenario, one could use a jigsaw puzzle task where patches are extracted from an image, scrambled randomly and then fed to a neural network that is trained to recover the original spatial ordering of the patches [10]. If the network performs this task reasonably well, then it is conceivable that it has learned some of the structure of natural images, and that the trained network could be reused as a feature extractor or weight initialization on a smaller-scale supervised learning problem such as object recognition. Apart from facilitating the downstream task and/or reducing the number of necessary annotated examples, self-supervision can also uncover more general and robust features than those learned in a specialized supervised task [11]. Therefore, given the potential benefits of SSL, can it be used to enhance the analysis of EEG?

To date, most applications of SSL have focused on domains where plentiful annotated data is already available, such as computer vision [9] and natural language processing [12, 13]. Particularly in computer vision, deep networks are often trained with fully supervised tasks (e.g., ImageNet pretraining). In this case, enough labeled data is available such that direct supervised learning on the downstream task is already in itself competitive [14]. SSL has an even greater potential in domains where low-labeled data regimes are common and supervised learning's effectiveness is limited, e.g. biosignal and EEG processing. Despite this, few studies on SSL and biosignals have been published. These studies either focus on limited downstream tasks and datasets [15], or test their approach on signals other than EEG [16].

Therefore, it still remains to be shown whether self-supervision can truly bring improvements over standard supervised approaches on EEG, and if this is the case, what the best ways of applying it are. Specifically, can we learn generic representations of EEG with self-supervision and, in doing so, reduce the need for costly EEG annotations? Given the growing popularity of deep learning as a processing tool for EEG [17], the answer could have a significant impact on current practices in the field of EEG processing. Indeed, while deep learning is notoriously data-hungry, an overwhelmingly large part of all neuroscience research happens in the low-labeled data regime, including EEG research: clinical studies with a few hundred subjects are often considered to be big data, while large-scale studies are much rarer and usually originate from research consortia [18, 19, 20, 21]. Therefore, it is to be expected that the performance reported by most deep learning-EEG studies - often in low-labeled data regimes - has so far remained limited and does not clearly outperform those of conventional approaches [17]. By leveraging unlabeled data, SSL can effectively create a lot more examples, which could enable more successful applications of deep learning to EEG.

In this paper, we investigate the use of self-supervision as a general approach to learning representations from EEG data. To the best of our knowledge, we present the first detailed analysis of SSL tasks on multiple types of EEG recordings. We aim to answer the following questions:

1. What are good SSL tasks that capture relevant structure in EEG data?
2. How do SSL features compare to other unsupervised and supervised approaches in terms of downstream classification performance?
3. What are the characteristics of the features learned by SSL? Specifically, can SSL capture physiologically- and clinically-relevant structure from unlabeled EEG?

The rest of the paper is structured as follows. Section 2 presents an overview of the SSL literature, then describes the different SSL tasks and learning problems considered in our study. We also introduce the neural architectures, baseline methods and data used in our experiments. Next, Section 3 reports the results of our experiments on EEG. Lastly, we discuss the results in Section 4.

2 Methods

2.1 State-of-the-art self-supervised learning approaches

Although it has not always been known as such, SSL has already been used in many other fields. In computer vision, multiple approaches have been proposed that rely on the spatial structure of images and the temporal structure of videos. For example, a context prediction task was used to train feature extractors on unlabeled images in [22] by predicting the position of a randomly sampled image patch relative to a second patch. Using this approach to pretrain a neural network, the authors reported improved performance as compared to a purely supervised model on the Pascal VOC object detection challenge. These results were among the first showing that self-supervised pretraining could help improve performance when limited annotated data is available. Similarly, the jigsaw puzzle task mentioned above [10] led to improved downstream performance on the same dataset. In the realm of video processing, approaches based on temporal structure have also been proposed: for instance, in [23], predicting whether a sequence of video frames were ordered or shuffled was used as a pretext task and tested on a human activity recognition downstream task. The interested reader can find other applications of SSL to images in [9].

Similarly, modern natural language processing (NLP) tasks often rely on self-supervision to learn word embeddings, which are at the core of many applications [24]. For instance, the original *word2vec* model was trained to predict the words around a center word or a center word based on the words around it [12], and then reused on a variety of downstream tasks [25]. More recently, a dual-task self-supervised approach, *BERT*, led to state-of-the-art performance on 11 NLP tasks such as question answering and named entity recognition [13]. The high performance achieved by this approach showcases the potential of SSL for learning general-purpose representations.

Lately, more general pretext tasks as well as improved methodology have led to strong results that have begun to rival purely supervised approaches. For instance, contrastive predictive coding (CPC), an autoregressive prediction task in latent space, was successfully used for images, text and speech [11]. Given an encoder and an autoregressive model, the task consists of predicting the output of the encoder for future windows (or image patches or words) given a context of multiple windows. The authors presented several improved results on various downstream tasks; a follow-up further showed that higher-capacity networks could improve downstream performance even more, especially in low-labeled data regimes [26]. Momentum contrast (MoCo), rather than proposing a new pretext task, is an improvement upon contrastive tasks, i.e., where a classifier must predict which of two or more inputs is the true sample [27, 28]. By improving the sampling of negative examples in contrastive tasks, MoCo helped boost the efficiency of SSL training as well as the quality of the representations learned. Similarly, it was found in [29] that using the right data augmentation transforms (e.g., random cropping and color distortion on images) and increasing batch size could lead to significant improvements in downstream performance.

The ability of SSL-trained features to demonstrably generalize to downstream tasks justifies a closer look at their statistical structure. A general and theoretically grounded approach was recently formalized by Hyvärinen *et al.* [30, 31] from the perspective of nonlinear independent components analysis. In this generalized framework, an observation $\mathbf{x}$ is embedded using an invertible neural network, and contrasted against an auxiliary variable $\mathbf{u}$ (e.g., the time index, the index of a segment or the history of the data). A discriminator classifies the pair by learning to predict whether $\mathbf{x}$ is paired with its corresponding auxiliary variable $\mathbf{u}$ or a perturbed (random) one $\mathbf{u}^*$. When the data exhibits certain structure (e.g., autocorrelation, non-stationarity, non-gaussianity), the embedder trained on this contrastive task will perform identifiable nonlinear ICA [31]. Most of the previously introduced SSL tasks can be viewed through this framework. Given the widespread use of linear ICA as a preprocessing and feature extraction tool in the EEG community [32, 33, 34, 35], an extension to the nonlinear regime is a natural step forward and could help improve traditional processing pipelines.

Remarkably, very few studies have applied SSL to biosignals despite its potential to leverage large quantities of unlabeled data. In [15], a model inspired by *word2vec*, called *wave2vec*, was developed to work with EEG and electrocardiography (ECG) time series. Representations were learned by predicting the features of neighbouring windows from the concatenation of time-frequency representations of EEG signals and demographic information. This approach was however only tested on a single EEG dataset and was not benchmarked against fully supervised deep learning approaches or expert feature classification. SSL has also been applied to ECG as a way to learn features for a downstream emotion recognition task: in [16], a transformation discrimination pretext task was used in which the model had to predict which transformations had been applied to the raw signal. While these results show the potential of self-supervised learning for biosignals, a more extensive analysis of SSL targeted at EEG is required to pave the way for practical applications.

2.2 Self-supervised learning pretext tasks for EEG

In this section, we describe the three SSL pretext tasks used in the paper. A visual explanation of the tasks can be found in Fig. 1.

Notation We denote by $[\![q]\!]$ the set $\{1, \ldots, q\}$ and by $[\![p, q]\!]$ the set $\{p, \ldots, q\}$ for any integer $p, q \in \mathbb{N}$. The index t refers to time indices in the multivariate time series $S \in \mathbb{R}^{C \times M}$, where M is the number of time samples and C is the dimension of samples (*e.g.,* channels). We assume for simplicity that each S has the same size. We denote by $y \in \{-1, 1\}$ a binary label used in the learning task.

2.2.1 Relative positioning

To produce labeled samples from the multivariate time series S, we propose to sample pairs of time windows $(x_t, x_{t'})$ where each window x_t, $x_{t'}$ is in $\mathbb{R}^{C \times T}$ and T is the duration of each window, and where the index t indicates the time sample at which the window starts in S. The first window x_t is referred to as the "anchor window". Our assumption is that an appropriate representation of the data should evolve slowly over time (akin to the driving hypothesis behind Slow Feature Analysis (SFA) [36, 37]) suggesting that time windows close in time should share the same label. In the context of sleep staging, for instance, sleep stages usually last between 1 to 40 minutes [38]; therefore, nearby windows likely come from the same sleep stage, whereas faraway windows likely come from different sleep stages. Given $\tau_{pos} \in \mathbb{N}$, which controls the duration of the positive context, and $\tau_{neg} \in \mathbb{N}$, which corresponds to the negative context around each window x_i, we sample N labeled pairs:

$$\mathcal{Z}_N = \{((x_{t_i}, x_{t'_i}), y_i) \mid i \in [\![N]\!], (t_i, t'_i) \in \mathcal{T}, y_i \in \mathcal{Y}\},$$

where $\mathcal{Y} = \{-1, 1\}$ and $\mathcal{T} = \{(t, t') \in [\![M - T + 1]\!]^2 \mid |t - t'| \leq \tau_{pos} \text{ or } |t - t'| > \tau_{neg}\}$. Intuitively, $\mathcal{T}$ is the set of all pairs of time indices (t, t') which can be constructed from windows of size T in a time series of size M, given the duration constraints imposed by the particular choices of τ_{pos} and τ_{neg}[1]. Here $y_i \in \mathcal{Y}$ is specified by the positive or negative contexts parameters:

$$y_i = \begin{cases} 1, & \text{if } |t_i - t'_i| \leq \tau_{pos} \\ -1, & \text{if } |t_i - t'_i| > \tau_{neg} \end{cases} \tag{1}$$

.

[1] The values of τ_{pos} and τ_{neg} can be selected based on prior knowledge of the signals and/or with a hyperparameter search.

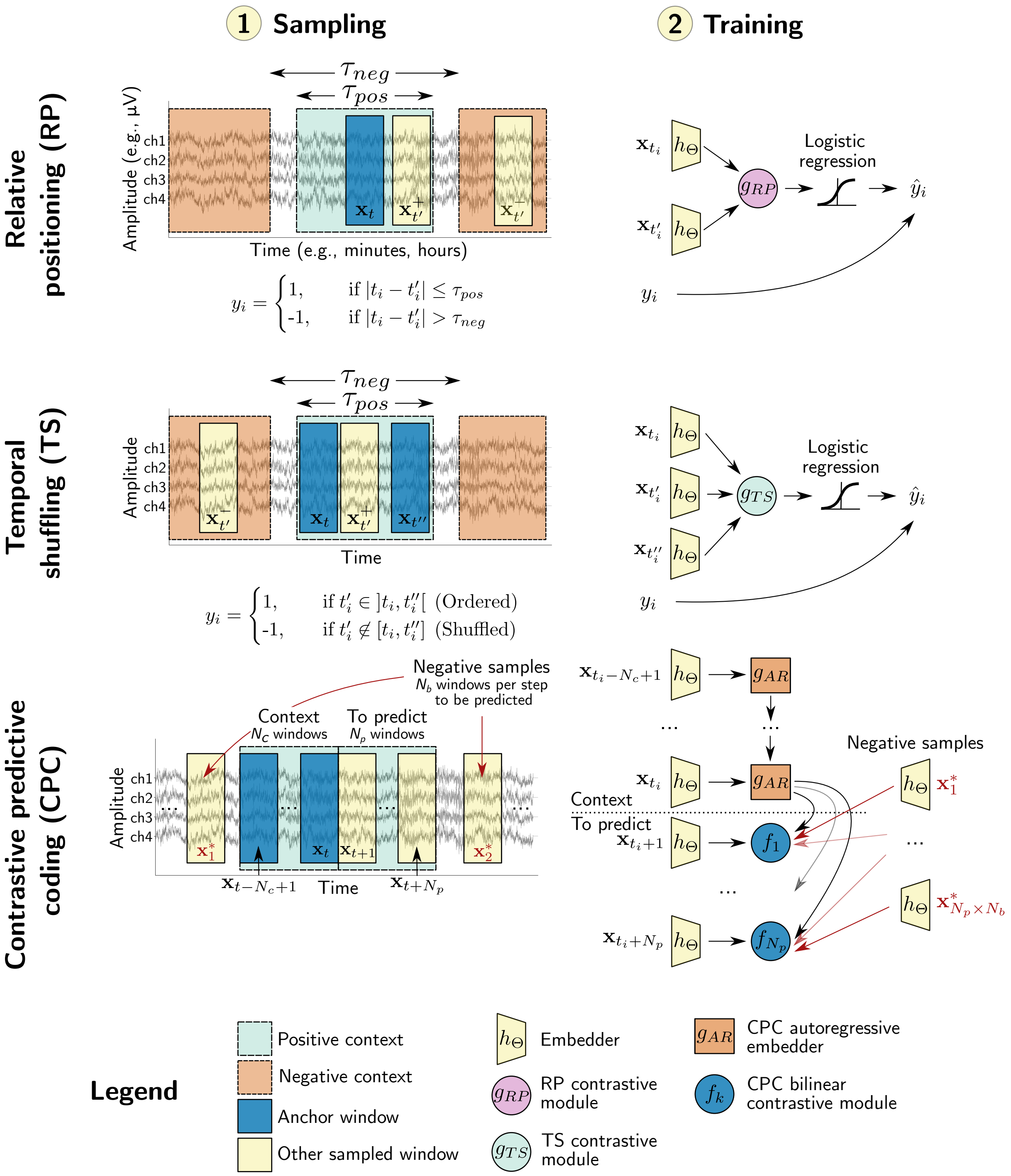

Figure 1: Visual explanation of the three SSL pretext tasks used in this study. The first column illustrates the sampling process by which examples are obtained in each pretext task. The second column describes the training process, where sampled examples are used to train a feature extractor h_Θ end-to-end.

We ignore window pairs where $x_{t'}$ falls outside of the positive and negative contexts of the anchor window x_t. In other words, the label indicates whether two time windows are closer together than τ_{pos} or farther apart than τ_{neg} in time. Noting the connection with the task in [22], we call this pretext task "relative positioning" (RP).

In order to learn end-to-end how to discriminate pairs of time windows based on their relative position, we introduce two functions h_Θ and g_{RP}. $h_\Theta : \mathbb{R}^{C\times T} \to \mathbb{R}^D$ is a feature extractor with parameters Θ which maps a window x to its representation in the feature space. Ultimately, we expect h_Θ to learn an informative representation of the raw EEG input which can be reused in different downstream tasks. A contrastive module g_{RP} is then used to aggregate the feature representations of each window. For the RP task, $g_{RP} : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}^D$ combines representations from pairs of windows by computing an elementwise absolute difference, denoted by the $|\cdot|$ operator: $g_{RP}(h_\Theta(x), h_\Theta(x')) = |h_\Theta(x) - h_\Theta(x')| \in \mathbb{R}^D$. The role of g_{RP} is to aggregate the feature vectors extracted by h_Θ on the two input windows and highlight their differences to simplify the contrastive task. Finally, a linear context discriminative model with coefficients $w \in \mathbb{R}^D$ and bias term $w_0 \in \mathbb{R}$ is responsible for predicting the associated target y. Using the binary logistic loss on the predictions of g_{RP} we can write a joint loss function $\mathcal{L}(\Theta, w, w_0)$ as

$$\mathcal{L}(\Theta, w, w_0) = \sum_{(x_t, x_{t'}, y) \in \mathcal{Z}_N} \log(1 + \exp(-y[w^\top g_{RP}(h_\Theta(x_t), h_\Theta(x_{t'})) + w_0])) \tag{2}$$

which we assume to be fully differentiable with respect to the parameters (Θ, w, w_0). Given the convention used for y, the predicted target is the sign of $w^\top g(h_\Theta(x_t), h_\Theta(x_{t'})) + w_0$.

2.2.2 Temporal shuffling

We also introduce a variation of the RP task that we call "temporal shuffling" (TS), in which we instead sample two anchor windows x_t and $x_{t''}$ from the positive context, and a third window $x_{t'}$ that is either between the first two windows or in the negative context. We then construct window triplets that are either temporally ordered ($t < t' < t''$) or shuffled ($t < t'' < t'$ or $t' < t < t''$). We augment the number of possible triplets by also considering the mirror image of the previous triplets, e.g., $(x_t, x_{t'}, x_{t''})$ becomes $(x_{t''}, x_{t'}, x_t)$. The label y_i then indicates whether the three windows are ordered or have been shuffled, similar to [23].

The contrastive module for TS is defined as $g_{TS} : \mathbb{R}^D \times \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}^{2D}$ and is implemented by concatenating the absolute differences:

$$g_{TS}(h_\Theta(x), h_\Theta(x'), h_\Theta(x'')) = (|h_\Theta(x) - h_\Theta(x')|, |h_\Theta(x') - h_\Theta(x'')|) \in \mathbb{R}^{2D} \ .$$

Moreover, Eq. (2) is extended to TS by replacing g_{RP} by g_{TS} and introducing $x_{t''}$ to obtain:

$$\mathcal{L}(\Theta, w, w_0) = \sum_{(x_t, x_{t'}, x_{t''}, y) \in \mathcal{Z}_N} \log(1 + \exp(-y[w^\top g_{TS}(h_\Theta(x_t), h_\Theta(x_{t'}), h_\Theta(x_{t''})) + w_0])) \ . \tag{3}$$

2.2.3 Contrastive predictive coding

The contrastive predictive coding (CPC) pretext task, introduced by Oord et al. [11], is defined here in comparison to RP and TS, as all three tasks share key similarities. Indeed, CPC can be seen as an extension of RP, where the single anchor window x_t is replaced by a sequence of N_c non-overlapping windows that are summarized by an autoregressive encoder $g_{AR} : \mathbb{R}^{D\times N_c} \to \mathbb{R}^{D_{AR}}$

with parameters Θ_{AR}[2]. This way, the information in the context can be represented by a single vector $c_t \in \mathbb{R}^{D_{AR}}$. g_{AR} can be implemented for example as a recurrent neural network with gated-recurrent units (GRU).

The context vector c_t is paired with not one, but N_p future windows (or "steps") which immediately follow the context. Negative windows are then sampled in a similar way as with RP and TS when $\tau_{neg} = 0$, i.e., the negative context is relaxed to include the entire time series. For each future window, N_b negative windows x^* are sampled inside each multivariate time series S ("same-recording negative sampling") or across all available S ("across-recording negative sampling"). For the sake of simplicity and to follow the notation of the original CPC article, we modify our notation slightly: we now denote a time window by x_t where t is the index of the window in the list of all non-overlapping windows of size T that can be extracted from a time series S. Therefore, the procedure for building a dataset with N examples boils down to sampling sequences X^c, X^p and X^n in the following manner:

$$\begin{aligned}
X_i^c &= (x_{t_i - N_c + 1}, \dots, x_{t_i}) && (N_c \text{ context windows})\\
X_i^p &= (x_{t_i + 1}, \dots, x_{t_i + N_p}) && (N_p \text{ future windows})\\
X_i^n &= (x_{t^*_{i_{1,1}}}, \dots, x_{t^*_{i_{1,N_b}}}, \dots, x_{t^*_{i_{N_p,1}}}, \dots, x_{t^*_{i_{N_p,N_b}}}) && (N_p N_b \text{ random negative windows})
\end{aligned}$$

where $t_i \in \llbracket N_c, M - N_p \rrbracket$. We denote with t^* time indices of windows sampled uniformly at random. The dataset then reads:

$$\mathcal{Z}_N = \{(X_i^c, X_i^p, X_i^n) \mid i \in \llbracket N \rrbracket\} \ . \tag{4}$$

As with RP and TS, the feature extractor h_Θ is used to extract a representation of size D from a window x_t. Finally, whereas the contrastive modules g_{RP} and g_{TS} explicitly relied on the absolute value of the difference between embeddings h, here for each future window x_{t+k} where $k \in \llbracket N_p \rrbracket$ a bilinear model f_k parametrized by $W_k \in \mathbb{R}^{D \times D_{AR}}$ is used to predict whether the window chronologically follows the context c_t or not:

$$f_k(c_t, h_\Theta(x_{t+k})) = h_\Theta(x_{t+k})^\top W_k c_t \tag{5}$$

The whole CPC model is trained end-to-end using the InfoNCE loss [11] (a categorical cross-entropy loss) defined as

$$\begin{aligned}
\mathcal{L}(\Theta, &\Theta_{AR}, W_k, \dots, W_{k+N_p-1}) = \\
&- \sum_{\substack{(X_i^c, X_i^p, X_i^n) \in \mathcal{Z}_N \\ c_{t_i} = g_{AR}(X_i^c)}} \sum_{k=1}^{N_p} \log \left[\frac{\exp(f_k(c_{t_i}, h_\Theta(x_{t_i+k})))}{\exp(f_k(c_{t_i}, h_\Theta(x_{t_i+k}))) + \sum_{j \in \llbracket N_b \rrbracket} \exp(f_k(c_{t_i}, h_\Theta(x_{t^*_{i_{k,j}}})))} \right]
\end{aligned} \tag{6}$$

While in RP and TS the model must predict *whether* a pair is positive or negative, in CPC the model must pick *which* of $N_b + 1$ windows actually follows the context. In practice, we sample batches of $N_b + 1$ sequences and for each sequence use the N_b other sequences in the batch to supply negative examples.

[2] CPC's encoder g_{AR} has parameters Θ_{AR}, however we omit them from the notation for brevity.

2.3 Downstream tasks

We performed empirical benchmarks of EEG-based SSL on two clinical problems that are representative of the current challenges in machine learning-based analysis of EEG: sleep monitoring and pathology screening. These two clinical problems commonly give rise to classification tasks, albeit with different numbers of classes and distinct data-generating mechanisms: sleep monitoring is concerned with biological events (event level) while pathology screening is concerned with single patients as compared to the population (subject level). These two clinical problems have generated considerable attention in the research community, which has led to the curation of large public databases. To enable fair comparisons with supervised approaches, we benchmarked SSL on the Physionet Challenge 2018 [1, 39] and the TUH Abnormal EEG [40] datasets.

First, we considered sleep staging, which is a critical component of a typical sleep monitoring assessment and is key to diagnosing and studying sleep disorders such as apnea and narcolepsy [41]. Sleep staging has been extensively studied in the machine (and deep) learning literature [42, 43, 17] (approximately 10% of reviewed papers in [17]), though not through the lens of SSL. Achieving fully automated sleep staging could have a substantial impact on clinical practice as (1) agreement between human raters is often limited [7] and (2) the annotation process is time-consuming and still largely manual [6]. Sleep staging typically gives rise to a 5-class classification problem where the possible predictions are W (wake), N1, N2, N3 (different levels of sleep) and R (rapid eye movement periods). Here, the task consists of predicting the sleep stages that correspond to 30-s windows of EEG.

Second, we applied SSL to pathology detection: EEG is routinely used in a clinical context to screen individuals for neurological conditions such as epilepsy and dementia [44, 45]. However, successful pathology detection requires highly specialized medical expertise and its quality depends on the expert's training and experience. Automated pathology detection could, therefore, have a major impact on clinical practice by facilitating neurological screening. This gives rise to classification tasks at the subject level where the challenge is to infer the patient's diagnosis or health status from the EEG recording. In the TUH dataset, medical specialists have labeled recordings as either pathological or non-pathological, giving rise to a binary classification problem. Importantly, these two labels reflect highly heterogeneous situations: a pathological recording could reflect anomalies due to various medical conditions, suggesting a rather complex data-generating mechanism. Again, various supervised approaches, some of them leveraging deep architectures, have addressed this task in the literature [46, 47, 48], although none has relied on self-supervision.

These two tasks are further described in Section 2.6 when discussing the data used in our experiments.

2.4 Deep learning architectures

We used two different deep learning architectures as embedders h_Θ in our experiments (see Fig. 2 for a detailed description). Both architectures were convolutional neural networks composed of spatial and temporal convolution layers, which respectively learned to perform the spatial and temporal filtering operations typical of EEG processing pipelines.

The first one, which we call StagerNet, was adapted from previous work on sleep staging where it was shown to perform well for window-wise classification of sleep stages [42]. StagerNet is a 3-layer convolutional neural network optimized to process windows of 30 s of multichannel EEG. As opposed to the original architecture, (1) we used twice as many convolutional channels

(16 instead of 8), (2) we added batch normalization after both temporal convolution layers[3] (3) we did not pad temporal convolutions and (4) we changed the dimensionality of the output layer to $D = 100$ instead of the number of classes (see Fig. 2-1). This yielded a total of 62,307 trainable parameters.

The second embedder architecture, ShallowNet, was directly taken from previous literature on the TUH Abnormal dataset [47, 48]. Originally designed to be a parametrized version of the filter bank common spatial patterns (FBCSP) processing pipeline common in brain-computer interfacing, ShallowNet has a single (split) convolutional layer followed by a squaring non-linearity, average pooling, a logarithm non-linearity, and a linear output layer. Batch normalization was used after the temporal convolution layer. Despite its simplicity, this architecture was shown in [48] to perform almost as well as the best model on the task of pathology detection on the TUH Abnormal dataset. We therefore used it as is, except for the dimensionality of the output layer which we also changed to $D = 100$ (See Fig. 2-2). This yielded a total of 170,860 trainable parameters.

We used a GRU with a hidden layer of size $D_{AR} = 100$ for the CPC task's g_{AR}, for experiments on both datasets.

The Adam optimizer [50] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and learning rate 5×10^{-4} was used. The batch size for all deep models was set to 256, except for CPC where it was set to 32. Training ran for at most 150 epochs or until the validation loss stopped decreasing for a period of a least 10 epochs (or 6 epochs for CPC). Dropout was applied to fully connected layers at a rate of 50% and a weight decay of 0.001 was applied to the trainable parameters of all layers. Finally, the parameters of all neural networks were randomly initialized using uniform He initialization [51].

2.5 Baselines

The SSL tasks were compared to four baseline approaches on the downstream tasks: (1) random weights, (2) convolutional autoencoders, (3) purely supervised learning and (4) handcrafted features.

The random weights baseline used an embedder whose weights were frozen after random initialization. The autoencoder (AE) was a more basic approach to representation learning, where a neural network made up of an encoder and a decoder learned an identity mapping between its input and its output, penalized by e.g., a mean squared error loss [52]. Here, we used h_Θ as the encoder and designed a convolutional decoder that inverts the operations of h_Θ. The purely supervised model was directly trained on the downstream classification problem, i.e., it had access to the labeled data. To do so, we added an additional linear classification layer to the embedder, before training the whole model with a multi-class cross-entropy loss.

Finally, we also included traditional machine learning baselines based on handcrafted features. For sleep staging, we extracted the following features [42]: mean, variance, skewness, kurtosis, standard deviation, frequency log-power bands between (0.5, 4.5, 8.5, 11.5, 15.5, 30) Hz as well as all their possible ratios, peak-to-peak amplitude, Hurst exponent, approximate entropy and Hjorth complexity. This resulted in 37 features per EEG channel, which were concatenated into a single vector. In the event of an artefact causing missing values in the feature vector of a window, we imputed missing values feature-wise using the mean of the feature computed over the training

[3] As described in [11, 27], batch normalization can harm the network's ability to learn on the CPC pretext task. However, we did not see this effect on our models (likely because their capacity is relatively small) and alternatives such as no normalization or layer normalization [49] performed unfavorably. Therefore, we also used batch normalization in CPC experiments.

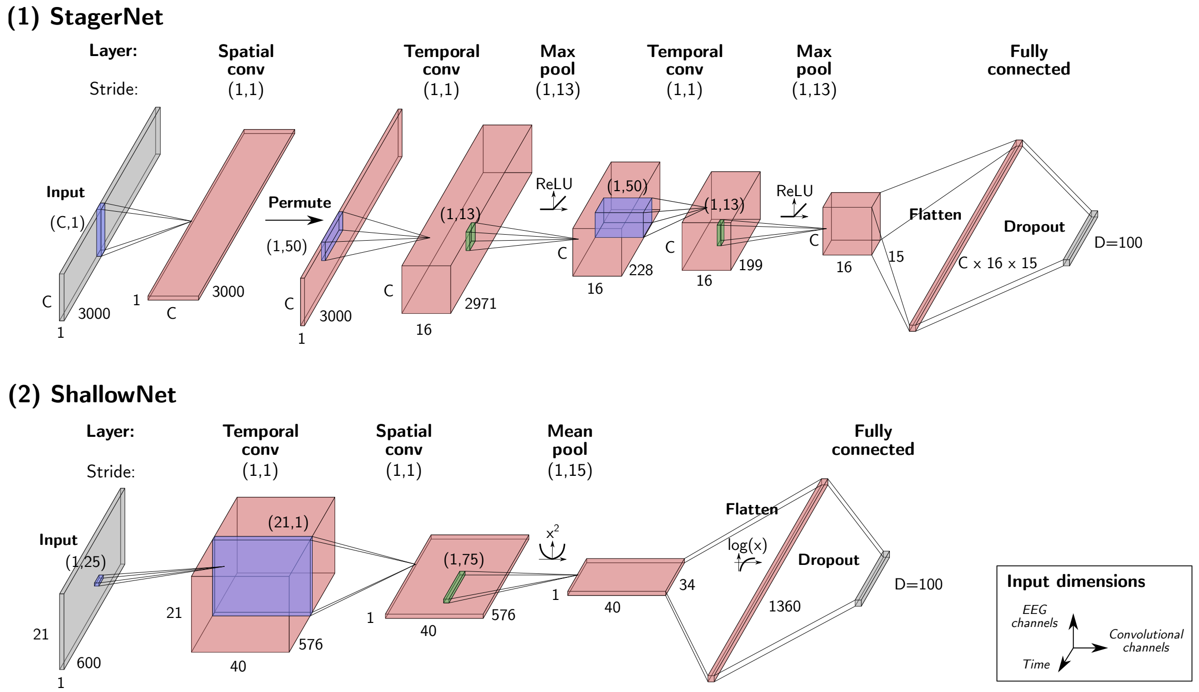

Figure 2: Neural network architectures used as embedder h_Θ for (1) sleep EEG and (2) pathology detection experiments.

set. For pathology detection, Riemannian geometry features were used like in [48], where it was reported that a nonlinear classifier trained on tangent space features reached high accuracy on the evaluation set of the TUH Abnormal dataset. We did not average the covariance matrices per recording to allow a fair comparison with the other methods which work window-wise. Therefore, for C channels of EEG, the input to the classifier had dimensionality $C(C+1)/2$.

For the downstream tasks, features learned with RP, TS, CPC and AE were classified using linear logistic regression with L2-regularization parameter[4] $C = 1$, while handcrafted features were classified using a random forest classifier with 300 trees, maximum depth of 15 and a maximum number of features per split of $\sqrt{F}$ (where F is the number of features)[5]. Balanced accuracy (bal acc), defined as the average per-class recall, was used to evaluate model performance on the downstream tasks. Moreover, during training, the loss was weighted to account for class imbalance. Models were trained using a combination of the braindecode [53], MNE-Python [54], pytorch [55], pyRiemann [56] and scikit-learn [57] packages. Finally, deep learning models were trained on 1 or 2 Nvidia Tesla V100 GPUs for anywhere from a few minutes to 7h, depending on the amount of data, early stopping and GPU configuration.

2.6 Data

The experiments were conducted on two publicly available EEG datasets, which are described in Tables 1 and 2.

[4] Varying C had little impact on downstream performance, and therefore we used a value of 1 across experiments.

[5] Random forest hyperparameters were selected using a grid search with maximum depth in $\{3, 5, 7, 9, 11, 13, 15\}$, and maximum number of features per tree in $\{\sqrt{F}, \log_2 F\}$ using the validation sets as described in Section 2.6.

PC18 (train)			
	# windows		
W	158,020	# unique subjects	994
N1	136,858	# recordings	994
N2	377,426	Sampling frequency	200 Hz
N3	102,492	# EEG channels	6
R	116,872	Reference	M1 or M2
Total	891,668		

Table 1: Description of the Physionet Challenge 2018 (PC18) dataset used in this study for sleep staging experiments.

TUHab				
	train	eval	# unique subjects	2329
	# recordings	# recordings	# recordings	2993
Normal	1371	150	Sampling frequency	250, 256, 512 Hz
Abnormal	1346	126	# EEG channels	27 to 36
Total	2717	276	Reference	Common average

Table 2: Description of the TUH Abnormal (TUHab) dataset used in this study for EEG pathology detection experiments.

2.6.1 Physionet Challenge 2018 dataset

First, we conducted sleep staging experiments on the Physionet Challenge 2018 (PC18) dataset [1, 39]. This dataset was initially released in the context of an open-source competition on the detection of arousals in sleep recordings, i.e., short moments of wakefulness during the night. A total of 1,983 different individuals with (suspected) sleep apnea were monitored overnight and their EEG, EOG, chin EMG, respiration airflow and oxygen saturation measured. Specifically, 6 EEG channels from the international 10/20 system were recorded at 200 Hz: F3-M2, F4-M1, C3-M2, C4-M1, O1-M2 and O2-M1. The recorded data was then annotated by 7 trained scorers following the AASM manual [58] into sleep stages (W, N1, N2, N3 and R). Moreover, 9 different types of arousal and 4 types of sleep apnea events were identified in the recordings. As the sleep stage annotations are only publicly available on about half the recordings (used as the training set during the competition), we focused our analysis on these 994 recordings. In this subset of the data, mean age is 55 years old (min: 18, max: 93) and 33% of participants are female.

2.6.2 TUH Abnormal EEG dataset

We used the TUH Abnormal EEG dataset v2.0.0 (TUHab) to conduct experiments on pathological EEG detection [40]. This dataset, a subset of [19], contains 2,993 recordings of 15 minutes or more from 2,329 different patients who underwent a clinical EEG in a hospital setting. Each recording was labeled as “normal” (1,385 recordings) or “abnormal” (998 recordings) based on detailed physician reports. Most recordings were sampled at 250 Hz (although some were sampled at 256 or 512 Hz) and contained between 27 and 36 electrodes. Moreover, the corpus is divided into a training and an evaluation set with 2,130 and 253 recordings each. The mean age across

	PC18 # recordings	RP/TS # tuples	CPC # sequences	**TUHab** # recordings	RP/TS # tuples	CPC # sequences
Train	595	1,190,000	877,792	2,171	868,400	642,144
Valid	199	398,000	294,272	543	217,200	160,224
Test	199	398,000	292,608	276	110,400	81,184
Total	993	1,986,000	1,464,672	2,990	1,196,000	883,552

Table 3: Number of recordings used in the training, validation and testing sets with PC18 and TUHab, as well as the number of examples for each pretext task.

all recordings is 49.3 years old (min: 1, max: 96) and 53.5% of recordings are of female patients.

2.6.3 Data splits and sampling

We split the available recordings from PC18 and TUHab into training, validation and testing sets such that the examples from each recording were only in one of the sets (see Table 3).

For PC18, we used a 60-20-20% random split, meaning there were 595, 199 and 199 recordings in the training, validation and testing sets respectively. For RP and TS, 2,000 pairs or triplets of windows were sampled from each recording. For CPC, the number of batches to extract from each recording was computed as 0.05 times the number of windows in that recording; moreover, we set the batch size to 32.

For TUHab, we used the provided evaluation set as the test set. The recordings of the development set were split 80-20% into a training and a validation set. Therefore, we used 2,171, 543 and 276 recordings in the training, validation and testing sets. Since the recordings were shorter for TUHab, we randomly sampled 400 RP pairs or TS triplets instead of 2000 from each recording. We used the same CPC sampling parameters as for PC18.

2.6.4 Preprocessing

The preprocessing of the EEG recordings differed for the two datasets. On PC18, the raw EEG was first filtered using a 30 Hz FIR lowpass filter with a Hamming window, to reject higher frequencies that are not critical for sleep staging [42, 59]. The EEG channels were then downsampled to 100 Hz to reduce the dimensionality of the input data. For the same reason, we focused our analysis on channels F3-M2 and F4-M1. Lastly, non-overlapping windows of 30 s of size (3000 x 2) were extracted.

On TUHab, a similar procedure to the one reported in [48] was used. The first minute of each recording was cropped to remove noisy data that occurs at the beginning of recordings. Longer files were also cropped such that a maximum of 20 minutes was used from each recording. Then, 21 channels that are common to all recordings were selected (Fp1, Fp2, F7, F8, F3, Fz, F4, A1, T3, C3, Cz, C4, T4, A2, T5, P3, Pz, P4, T6, O1 and O2). EEG channels were downsampled to 100 Hz and clipped at $\pm 800\,\mu V$ to mitigate the effect of large artifactual deflections in the raw data. Non-overlapping 6-s windows were extracted, yielding windows of size (600×21).

Finally, windows from both datasets with peak-to-peak amplitude below $1\,\mu V$ were rejected. The remaining windows were normalized channel-wise to have zero-mean and unit standard deviation.

3 Results

We investigated the use of SSL tasks to learn useful EEG features from unlabeled data in a series of three experiments. First, SSL approaches were compared to fully supervised approaches based on deep learning or handcrafted features. Second, we explored SSL-learned representations to highlight clinically-relevant structure. Finally, in the last experiment, we studied the impact of hyperparameter selection on pretext and downstream performance.

3.1 SSL models learn representations of EEG and facilitate downstream tasks with limited annotated data

Can the suggested pretext tasks enable SSL on clinical EEG data and mitigate the amount of labeled EEG data that is required in clinical tasks? To address this question, we applied the pretext tasks to two clinical datasets (PC18 and TUHab) and compared their downstream performance to the one of various established approaches such as fully supervised learning, while varying the number of labeled examples available.

Context and setup. Feature extractors h_Θ were trained using the different approaches (AE, RP, TS and CPC on unlabeled data) and then used to extract features. Following hyperparameter search (see Section 3.3), we used same-recording negative sampling on PC18 and across-recording negative sampling on TUHab. We also extracted features with randomly initialized models. Downstream task performance was then evaluated by training linear logistic regression models on labeled examples, where the training set contains at least one and up to all existing labeled examples. Additionally, fully supervised models were trained directly on labeled data and random forests were trained on handcrafted features.

The impact of the number of labeled samples on downstream performance is presented in Fig. 3. First, when using SSL-learned features for the downstream tasks, we observe important above-chance performance across all data regimes: on PC18, our models scored as high as 72.3% balanced accuracy (5-class, chance=20%) while on TUHab the highest performance was of 79.4% (2-class, chance=50%). These results demonstrate the ability of SSL to learn useful representations for our downstream tasks. Second, the comparison suggests that SSL-learned features are competitive with other baseline approaches and can even outperform supervised approaches. On the PC18 sleep data (Fig. 3**A**), one can observe that all three SSL pretext tasks outperformed alternative approaches including the fully supervised model and handcrafted features in most data regimes. The performance gap between SSL-learned features and full supervision was as high as 22.8 points when only one example per class was available. It remained in favor of SSL up to around 10,000 examples per class, where full supervision finally began to exceed SSL performance, however by a 1.6-3.5% margin only. Moreover, SSL outperformed the handcrafted features baseline over 100 examples per class, e.g., by up to 5.6 points for CPC. These results suggest two important implications: (1) our pretext tasks can capture critical information for sleep staging, even though no sleep labels were available when learning representations and (2) these features can rival both human-engineered features and label-intensive full supervision.

Other baselines such as random weights and autoencoding obtained much lower performance, showing that learning informative features for sleep staging is not trivial and requires more sophistication than the inductive bias of a convolutional neural network alone or a pure reconstruction task. Interestingly, the poor performance of the AE can be attributed to its mean squared error loss. This encourages the model to focus on the signal's low frequencies, which, due to 1/f power-law dynamics have the largest amplitudes in biosignals like EEG. Yet, low frequency

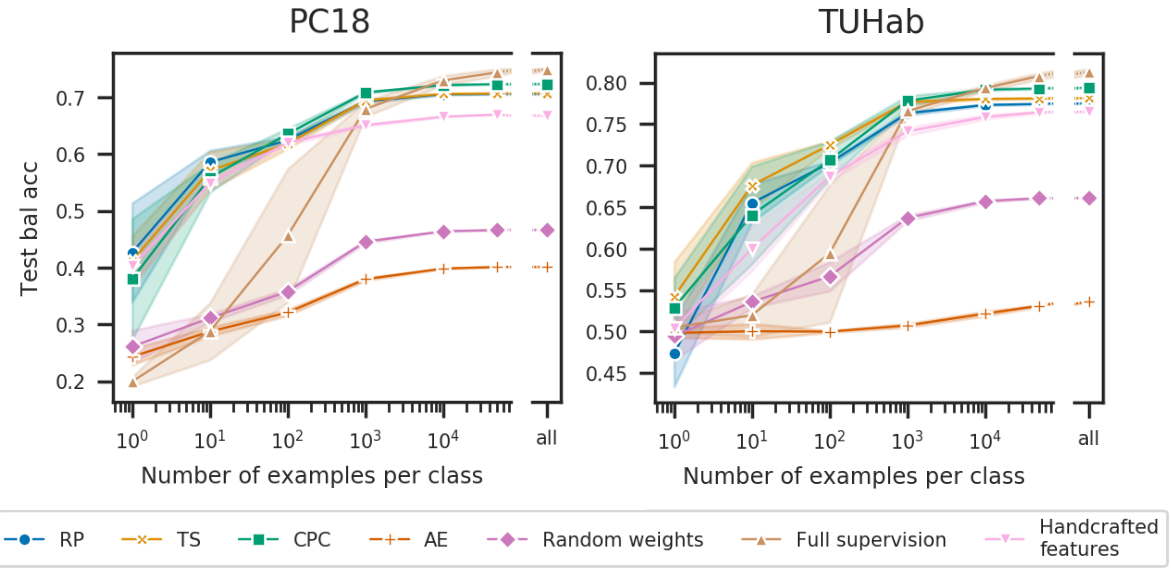

Figure 3: Impact of number of labeled examples per class on downstream performance. Feature extractors were trained with an autoencoder (AE), the relative positioning (RP), temporal shuffling (TS) and contrastive predictive coding (CPC) tasks, a fully supervised model, a model with random weights and handcrafted features, on PC18 and on TUHab. "All" means all available training examples were used. The same-recording negative sampling strategy was used for SSL tasks on PC18, while the across-recording strategy was used on TUHab. Results are the average of five runs with same initialization but different random selection of examples and shaded area represents standard deviation. While higher numbers of labeled examples led to better performance, SSL models achieved much higher performance than a fully supervised model when few labeled examples were available.

signals only capture a small portion of the neurobiological information in EEG signals.

Next, we applied SSL to the task of pathology detection, where the two classes ("normal" and "abnormal") are likely to be more heterogenous than the sleep staging classes. Again, SSL-learned features outperformed the baseline approaches in most data regimes: CPC outperformed full supervision when fewer than 10,000 labeled examples per class were available, while the performance gap between the two methods was on the order of 1% when all examples were available. Handcrafted features were also consistently outperformed by RP, TS and CPC, albeit by a smaller amount (e.g., 3.8-4.8 point difference for CPC). Again, the AE and random weights features could not compete with the other methods. Notably, the AE fared even worse on TUHab and downstream performance never exceeded 53.0%.

Taken together, our results demonstrate that the proposed SSL pretext tasks were general enough to enable two fundamentally different types of EEG classification problems. All SSL tasks systematically outperformed or equaled other approaches in low-to-medium labeled data regimes and remained competitive in a high labeled data regime.

3.2 SSL models capture physiologically and clinically meaningful features

While SSL-learned features yielded competitive performance on sleep staging and pathology detection tasks, it is unclear what kind of structure was captured by SSL. To address this, we

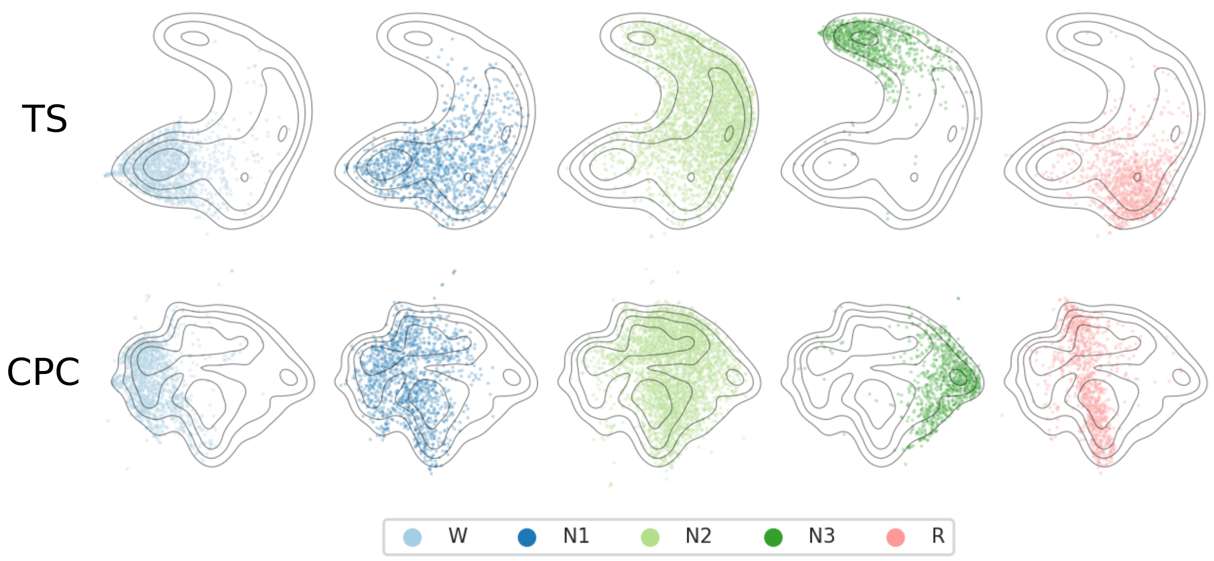

Figure 4: UMAP visualization of SSL features on the PC18 dataset. The subplots show the distribution of the 5 sleep stages as scatterplots for TS (first row) and CPC (second row) features. Contour lines correspond to the density levels of the distribution across all stages and are used as visual reference. Finally, each point corresponds to the features extracted from a 30-s window of EEG. In both cases, there is clear structure related to sleep stages although no labels were available during training.

examined the embeddings by analyzing their relationship with the different annotations and meta-data available in clinical datasets. We thus projected the 100-dimensional embeddings obtained on PC18 and TUHab onto a two-dimensional representation using Uniform Manifold Approximation and Projection (UMAP) [60] and using the best models as identified in Section 3.3. This allows a qualitative analysis of the local and global structure of the SSL-learned features.

Results on sleep data are shown in Fig. 4. A structure that closely follows the different sleep stages can be noticed in the embeddings of PC18 obtained with TS and CPC[6]. Upon inspection of the distribution of examples from the different stages, clear groups emerged. They not only corresponded to the labeled sleep stages, but they are also sequentially arranged: moving from one end of the embedding to another, we can draw a trajectory that passes through W, N1, N2 and N3 sequentially. Stage R, finally, mostly overlaps with N1. These results are in line with previous observations on the structure of the sleep-wakefulness continuum [61, 62].

Intuitively, the largest sources of variation in sleep EEG data are linked to changes in sleep stages and the corresponding microstructure (e.g., slow waves, sleep spindles, etc.). Can we expect other sources of variation to also be visible in the embeddings? To address this question, we inspected clinical information available in PC18 along with the embeddings: apnea events and subject age. The results are presented in Fig. 5. First, apnea-related structure can be seen in the middle of the embeddings, overlapping with the area where stage N2 was prevalent (first column of Fig. 5). At the same time, very few apnea events occurred at the extremities of the embedding, for instance over W regions, naturally, but also over N3 regions. Although this structure likely reflects the correlation between sleep stages, age and actual apnea-induced EEG patterns, this

[6] Results for RP were similar to TS's and are presented in Appendix.

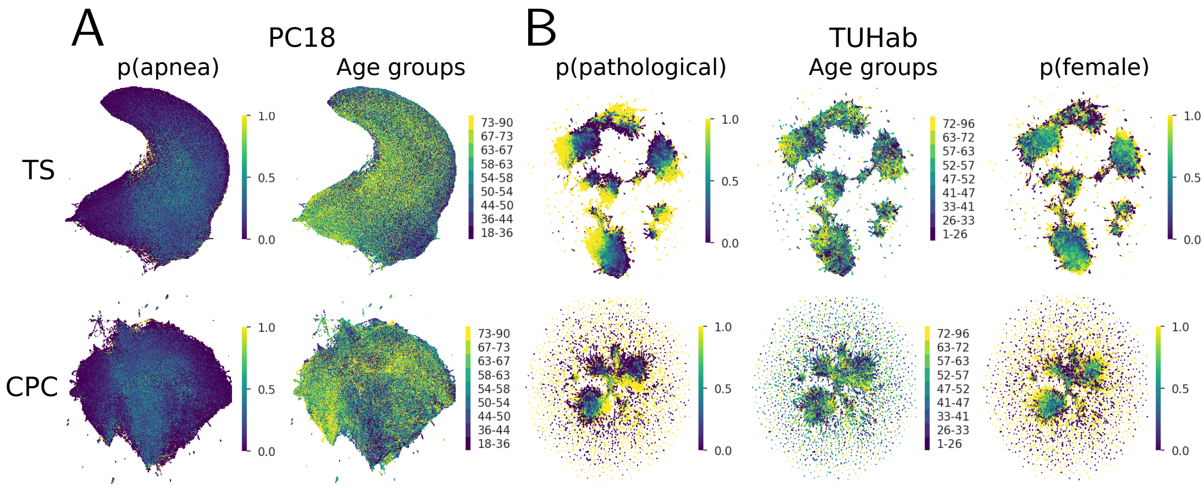

Figure 5: Structure learned by the embedders trained on SSL tasks. The 2D embedding spaces obtained with UMAP on (**A**) PC18 and (**B**) TUHab were discretized into 500 x 500 "pixels". For binary labels ("apnea", "pathological" and "gender"), we visualize the probability as heatmaps, i.e., the color indicates the probability that the label is true (e.g., that a window in that region of the embedding overlaps with an apnea annotation). For age, the subjects of each dataset were divided into 9 quantiles, and the color indicates which group was the most frequent in each bin. The features learned with the SSL tasks capture physiologically-relevant structure, such as pathology, age, apnea and gender.

nonetheless shows the potential of SSL to learn features that relate to clinical phenomena. Second, age structure was revelead in at least two distinct ways in the embeddings (second column of Fig. 5). The first is related to sleep macrostructure, i.e., the sequence of sleep stages and their relative length. Indeed, younger subjects were predominant over the R stage region, while older subjects were more frequently found over the W region. This is in line with well-documented phenomena such as increased sleep fragmentation and sleep onset latency in older individuals, as well as a subtle reduction in REM sleep with age [63]. Concurrently, we also observe sleep microstructure in the embeddings. For instance, looking at N2-N3 regions for the TS embedding, older age groups are more likely to be found in the leftmost side of the blob, while younger subjects are more likely to be found on its rightmost side. This suggests there is a difference between the characteristics of N2-N3 sleep across age groups, e.g., related to sleep spindles [64].

Can this clinically-relevant structure also be learned on a different type of EEG recording? We conducted a similar analysis for TUHab, this time focusing on pathology, age and gender. Results are shown in columns 3-5 of Fig. 5B. The embeddings of both TS and CPC exhibited a primary multi-cluster structure, with similar gradient-like structure inside each cluster[7]. For instance, pathology-related structure was clearly exposed in the two embeddings (column 3), with an increasing probability of the EEG being abnormal when moving from one end of the different clusters to the other. Likewise, an age-related gradient emerged inside each cluster (column 4), in a similar direction as the pathology gradient, while there is also a gender-associated gradient that appeared orthogonal to the first two (last column). What do the different clusters actually represent? We plotted experimental setup-related labels (the original number of EEG

[7] Results for RP are presented in Appendix.

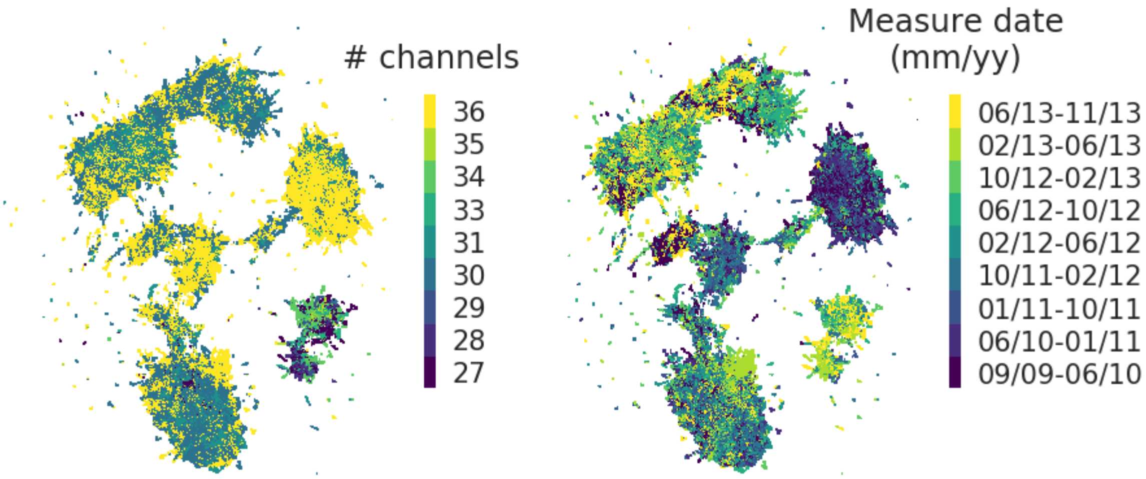

Figure 6: Structure related to the original recording's number of EEG channels and measurement date in TS-learned features on the entire TUHab dataset. The overall different number of EEG channels and measurement date in each cluster shows that the cluster-like structure reflects differences in experimental setups. See Fig. 5 for a description of how the density plots are computed.

channels and the measurement date of each recording) in Fig. 6. Each cluster was predominantly composed of examples with a given number of channels and with a specific range of measurement dates. This might suggest that the SSL tasks have partially learned the noise introduced by data collection. For example, the TUHab dataset was collected over many years across different sites, by different EEG technicians and with various EEG devices. Most likely, the impact of this noise in the embedding could be mitigated by using more aggressive preprocessing (e.g., bandpass filtering) or by sampling negative examples within recordings from the same cohort.

In conclusion, this experiment showed that SSL can encode clinically-relevant structure such as sleep stages, pathology, age, apnea and gender information from EEG data, while revealing interactions (such as young age and REM sleep), without any access to labels.

3.3 SSL pretext task hyperparameters strongly influence downstream task performance

How should the various SSL pretext task hyperparameters be tuned to fully make use of self-supervision in clinical EEG tasks? In this section, we describe how the hyperparameters of the models used in the experiments above were tuned and study in detail the impact of some key hyperparameters on downstream performance.

To benchmark different pretext tasks across datasets, we tracked the performance of the pretext and downstream tasks across different choices of hyperparameters (see Section B for a complete description of the search procedure). The comparison is depicted in Fig. 7. The analysis suggests that the pretext tasks performed significantly above chance level on all datasets: RP and TS reached a maximum performance of 98.0% (2-class, chance=50%) while CPC yielded performances as high as 95.4% (32-class, chance= 3.1%). On the downstream tasks, SSL-learned representations

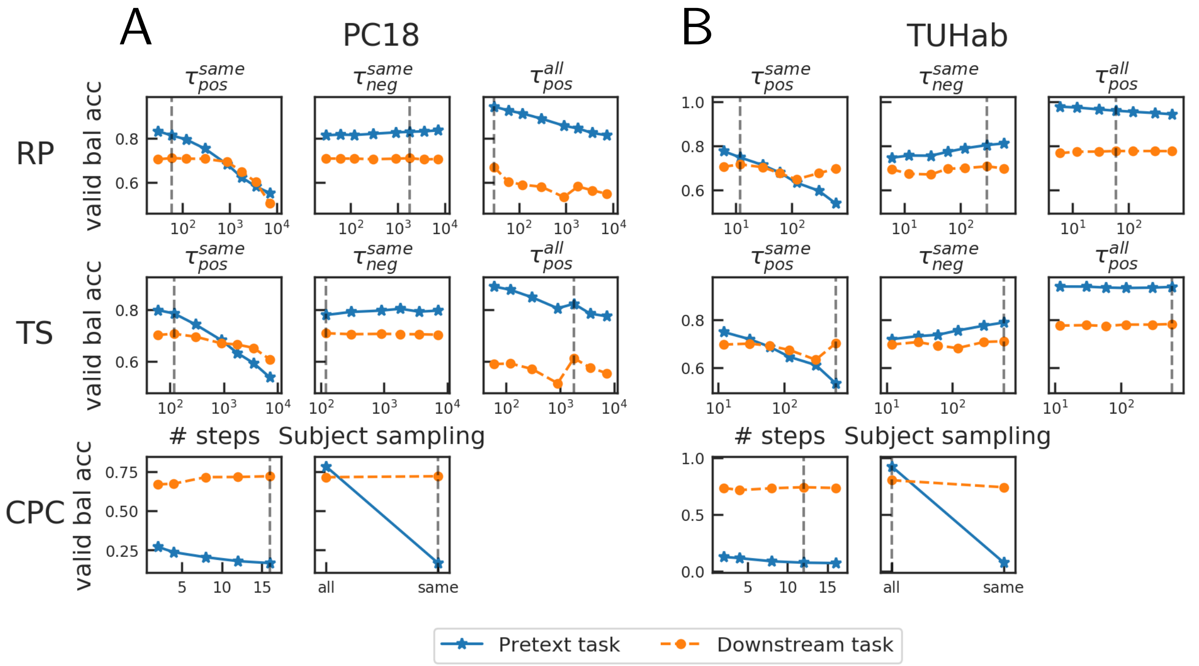

Figure 7: Impact of principal hyperparameters on pretext (blue) and downstream (orange) task performance, measured with balanced accuracy on the validation set on (A) PC18 and (B) TUHab. Each row corresponds to a different SSL pretext task. For both RP and TS, we varied the hyperparameters that control the length of the positive and negative contexts (τ_{pos}, τ_{neg}, in seconds); the exponent "same" or "all" indicates whether negative windows were sampled across the same recording or across all recordings, respectively. For CPC, we varied the number of predicted windows and the type of negative sampling. Finally, the best hyperparameter values in terms of downstream task performance are emphasized using vertical dashed lines. See text for more details on the hyperparameter search procedure.

always performed above chance as reported in Section 3.1. Interestingly though, configurations with high pretext performance did not necessarily lead to high downstream performance, which highlights the necessity of appropriate hyperparameter selection.

In the next step, we examined the influence of the different hyperparameters on each pretext task (rows of Fig. 7) to identify optimal configurations. First, we focused on the same-recording negative sampling scenario, in which negative examples are sampled from the same recording as the anchor window(s). With RP, increasing τ_{pos} always made the pretext task harder. This is expected since the larger the positive context, the more probable it is to get positive example pairs that are composed of distant (and thus potentially dissimilar) windows. On sleep data, we notice a plateau effect: the downstream performance was more or less constant below $\tau_{pos} = 20$ min, suggesting EEG autocorrelation properties might be changing at this temporal scale. Although this phenomenon did not appear clearly on TUHab, downstream performance decreased above $\tau_{pos} = 30$ s, and then increased again after $\tau_{pos} = 2$ min. On the other hand, varying τ_{neg} given a fixed τ_{pos} did not have a consistent or significant influence on downstream performance, although larger τ_{neg} generally led to easier pretext tasks.

Do these results hold when negative windows are sampled across all recordings? Interestingly, the type of negative sampling has a considerable effect on downstream performance (columns 3 and 6 of Fig. 7). On sleep staging, downstream performance dropped significantly and degraded faster as τ_{pos} was increased, while the opposite effect could be seen on the pathology detection task (higher, more stable performance). This effect might be explained by the nature of the downstream task: in sleep staging, major changes in the EEG occur inside a given recording, therefore distinguishing between windows of a same recording is key to identifying sleep stages. On the other hand, in pathology detection, each recording is given a single label ("normal" or "pathological") and so being able to know whether a window comes from the same recording (necessarily with the same label) or from another one (possibly with the opposite label) intuitively appears more useful. In other words, the distribution that is chosen for sampling negative examples determines the kind of invariance that the network is forced to learn. Overall, similar results hold for TS.

As for CPC, a similar analysis shows that while increasing the number of windows to predict ("number of steps") made the pretext task more difficult, predicting further ahead in the future helped the embedder learn a better representation for sleep staging (bottom row of Fig. 7). Were these results affected by the type of negative sampling as was the case for RP and TS? Remarkably, the type of negative sampling had a minor effect on downstream performance on sleep data (71.6 vs. 72.2% bal acc), but had a considerable effect on pathology detection (74.1 vs. 80.4%), akin to what was seen above on RP and TS. This result echoes the report of [11] where subject-specific negative sampling led to the highest downstream performance on a phoneme classification downstream task.

In this last experiment, we confirmed that our SSL pretext tasks are not trivial, and that certain pretext task hyperparameters have a measurable impact on downstream performance.

4 Discussion

In this paper, we introduced self-supervised learning (SSL) as a way to learn representations on EEG data. Specifically, we proposed two SSL tasks designed to capture structure in EEG data, relative positioning (RP) and temporal shuffling (TS) and adapted a third approach, contrastive predictive coding (CPC), to work on EEG data. We showed that these tasks can be used to learn generic features which capture clinically relevant structure from unlabeled EEG, such as sleep micro- and macrostructure and pathology. Moreover, we performed a rigorous comparison of SSL methods to traditional unsupervised and supervised methods on EEG, and showed that downstream classification performance can be significantly improved by using SSL, particularly in low-labeled data regimes. These results hold for two large-scale EEG datasets comprising sleep and pathological EEG data, both with thousands of recordings.

4.1 Using SSL to improve performance in semi-supervised scenarios

We showed that SSL can be used to improve downstream performance when a lot of unlabeled data is available but the amount of labeled examples is limited, i.e., in a semi-supervised learning scenario (see Section 4.1). For instance, CPC-learned features outperformed fully supervised learning on sleep data by about 20% when only one labeled example per class was available. Similarly, on the pathology detection task an improvement of close to 15% was obtained with SSL when only 10 labeled examples per class were available. In practice, SSL has the potential to become a common tool for boosting classification performance when annotations are expensive, a common scenario when working with biosignals like EEG.

Can SSL be applied to any semi-supervised problem or is it limited to larger clinical datasets? While the SSL pretext tasks included in this work are applicable to multivariate time series in general, their successful application does require adequate recording length and dataset size. First, the EEG recordings need to be sufficiently long so that a reasonable number of windows can be sampled given the positive and negative contexts and the window length. For clinical recordings, this is typically not an issue: sleep monitoring and pathology screening procedures both produce recordings of tens of minutes to many hours, which largely suffice. It is noteworthy that while the proposed SSL approaches have been developed using clinical data, they could be readily applied to event-related EEG protocols, such as those encountered in cognitive psychology or brain-computer interfacing experiments [65, 66]. Indeed, SSL could be applied to an entire recording without explicitly taking into consideration the known events (e.g. stimuli, behavioral responses). This critically depends on the availability of continuous recordings (rather than epoched data only, as is the case in some public EEG datasets) including resting state baselines and between-trial segments. Second, the current results may suggest large datasets are necessary to enable SSL on EEG as analyses were based on two of the largest publicly available EEG datasets with thousands of recordings each. However, similar results hold on much smaller datasets containing fewer than 100 recordings [67]. Intuitively, as long as the unlabeled dataset is representative of the variability of the test data, representations that are useful for the pretext task should be transferable to a related downstream task.

One might argue that the observed performance benefit of SSL is incremental as compared to supervised approaches when the number of samples is moderate to large. Is investing in the development of SSL-based EEG-analysis worth the effort? It is important to highlight that our results present a proof of concept that opens the door to further developments, which may lead to substantial improvements in performance. In this paper, we limited our experiments to the linear evaluation protocol, where the downstream task is carried out by a linear classifier trained on SSL-learned features, in order to focus on the properties of the learned representations. Finetuning the parameters of the embedders on the downstream task [11, 68] could likely further improve downstream performance. Preliminary experiments (not shown) suggested that a 3 to 4-point improvement can be obtained in some data regimes when finetuning the embedders and that performance with all data points is as high as with a purely supervised model. However, using nonlinear classifiers (here, random forests) as suggested in [11] on SSL-learned features did not improve results, suggesting our downstream tasks might be sufficiently close to the pretext task that the relevant information is already linearly accessible. Another potential opportunity to improve downstream performance consists of using larger deep learning architectures. Due to their sampling methodology, most SSL tasks can "create" a vast number of distinct examples going far beyond the number of labeled examples typically available in a supervised task. For instance, on PC18, our training set contained close to 5×10^5 labeled examples while for our self-supervised tasks we chose to train our embedders on more than twice that number of examples (to fit computational time requirements), though more could have been easily sampled. The much higher number of available examples in SSL opens the door to using much larger deep neural networks which require much larger sample sizes [26]. Given the relatively shallow architectures currently used in deep learning and EEG research (on which we based our choice of architectures) [17], SSL could be key to training deeper models and improving upon current state of the art on various EEG tasks.

4.2 Sleep-wakefulness continuum and inter-rater reliability

We have demonstrated that the embeddings learned with SSL capture clinically-relevant information. Sleep, pathology, age and gender information appeared as clear structure in the learned feature space (see Fig. 4-5). The variety of metadata that is visible in the embeddings highlights the capacity of the proposed SSL tasks to uncover important factors of variation in noisy biosignal data such as EEG in a purely unsupervised manner. Critically though, this structure is not discrete, but continuous. Indeed, sleep stages are not cleanly separated into five clusters (or two clusters for normal and abnormal EEG), but instead the embeddings display a smooth continuum of sleep-wakefulness (or of normal-abnormal EEG). Is this gradient-like structure meaningful, or is it a mere artefact of our experimental setup? We argue that the continuous nature of SSL-learned features is inherent to the neurophysiological phenomena under study. Conveniently, this offers interesting opportunities to improve the analysis of physiological data.

While sleep EEG is routinely divided into discrete stages to be analyzed, its true nature is likely continuous. For instance, the concept of sleep stages and the taxonomy that we know today is the product of incremental standardization efforts in the sleep research community [69, 70, 71, 58]. Although this categorization is convenient, critics still stress the limitations of stage-based systems under the evidence of sub-stages of sleep and the interplay between micro- and macrostructure [72]. Moreover, even trained experts using the same set of rules do not perfectly agree on their predictions, showing that the definition of stages remains ambiguous: in [7], an overall agreement of 82.6% was obtained between the predictions of more than 2,500 trained sleep scorers (even lower for N1, at 63.0%). Consequently, could a better representation of sleep EEG, such as the one learned with self-supervision, alleviate some of these challenges? While previous research suggests sleep might indeed be measured using a continuous metric derived from a supervised machine learning model [61] or a computational mean-field model [62], we additionally demonstrated that the rich feature space learned with SSL can simultaneously capture sleep-related structure and variability caused by age and apnea. Importantly, the data-driven nature of the SSL representation alleviates the subjectivity of manual sleep staging. Overall, this suggests SSL-learned representations could provide more fine-grained information about the multiple factors at play during sleep and, in doing so, enable a more precise study of sleep.

Similarly, many EEG pathologies are described by a clinical classification system which defines discrete subtypes of diseases or disorders, e.g., epilepsy [73, 74] and dementia [75]. As for sleep EEG, inter-rater agreement is limited [48]. This suggests that there is an opportunity for these pathologies to be interpreted as a continuum as well. Although our pathology detection downstream task was a simple binary classification task, the clear pathology-associated gradient captured by our SSL pretext tasks could be used to characterize the different types and subtypes of pathologies contained in the dataset more precisely (see Fig. 5). Ultimately, the data-driven feature space obtained with SSL might aid in the study and comparison of different neuropathologies and EEG patterns in general.

4.3 Finding the right pretext task for EEG

With the large number of self-supervised pretext tasks one can think of, and the even larger number of possible EEG downstream tasks, how can we choose a combination of pretext task and hyperparameters for a given setting? To answer this question, many more empirical experiments will have to be conducted on EEG data. However, the results presented here give some insight as to what may be important to consider. In this work, we developed pretext tasks that proved effective on two different classification problems by using a combination of (1) prior knowledge about EEG time series, (2) assumptions about the statistical structure of the features to be learned, (3) thorough hyperparameter search and (4) computational considerations.

First, we introduced and tailored to EEG the relative positioning (RP) and temporal shuffling (TS) tasks by relying on prior knowledge about EEG. These tasks were designed specifically with the structure of sleep data in mind. Indeed, sleep EEG signals have a clear temporal structure originating from the succession of sleep stages during the night. This means that two windows that are close in time have a high probability of sharing the same sleep stage annotation and statistical structure. Therefore, learning to differentiate close-by from faraway windows should intuitively be related to learning to differentiate sleep stages. Similar approaches were previously described in the computer vision literature [22, 23], however they rely on properties of natural images that generally do not hold for EEG. For instance, whereas two EEG windows x_t and $x_{t'}$ that are close in time likely look alike, there is typically no physiological information in these windows that would allow one to determine whether $t < t'$ or $t > t'$.[8] Therefore, tasks that rely on proximity rather than absolute positioning appear to be a better match for EEG. As for CPC, we included it in our experiments as it is a natural extension of RP and TS and has led to promising results on other kinds of data [11].

Second, assumptions about the statistical structure of the latent factors or features to recover was used to further support our choice of tasks. For instance, given its similarity with permutation contrastive learning (PCL, a self-supervised method for performing nonlinear ICA [30]), RP likely relies on the general temporal dependencies (including autocorrelations) of EEG signals to recover informative features.[9] Since TS and CPC can both be seen as extensions of the RP task with more elaborate sampling strategies and contrastive procedures (see Section 2.2), all three tasks might effectively rely on similar structure to discover features.

Third, the careful selection of pretext task hyperparameters was essential to selecting the right pretext task configuration. For instance, RP, TS and CPC often yielded very similar downstream task performance once the best hyperparameters were selected. Out of the main pretext task hyperparameters, the negative sampling strategy proved to be especially important to tune (Section 3.3). Indeed, sleep staging benefited from same-recording negative sampling whereas pathology detection instead worked better when across-recording negative sampling was used. Interestingly, this appears to be the unique change to RP, TS and CPC that is required to compete with purely supervised approaches on the pathology detection downstream task, although RP and TS were initially designed for capturing intra-recording sleep structure. Thus, negative sampling hyperparameters might be among the most critical hyperparameters to tune, as they can be used to develop invariances to particular structure that is not desirable, e.g., intra-recording changes or measurement-site effects (Fig. 6). Ultimately, the fact that all three pretext tasks could reach similar downstream performance suggests self-supervision was able to uncover fundamental information likely related to physiology.

Finally, computational requirements and architecture-specific constraints are important to consider when choosing a pretext task. For example, being simpler, RP and TS might be preferred over CPC if they can reach similar performance. Indeed, CPC has more hyperparameters than RP

[8] Exceptions would include transitions between sleep stages that are more likely than others, such as from lighter to deeper sleep stages; however these transitions occur rarely during the night; more often a back-and-forth between sleep stages is observed.

[9] PCL can be obtained by setting RP's τ_{pos} to be the length of a single window and τ_{neg} to 0. Incidentally, we found that the optimal value of τ_{pos} and τ_{neg} were relatively small on the datasets considered, suggesting hyperparameters close to those of PCL are optimal.

and TS (i.e., number of context windows, of windows to be predicted and of negative examples; architecture of the autoregressive embedder g_{AR}) and requires additional forward and backward passes through the embedder h_Θ at training time. However, CPC's autoregressive encoder g_{AR} could yield better features for some tasks with larger-scale dependencies [11], e.g., sleep staging: indeed, previous studies on deep learning for automated polysomnography have reported improved performance using multiple consecutive windows [42] or a recurrent network on top of a window encoder [76]. This aggregation of window-level information is already part of a CPC model and therefore the pretrained g_{AR} could be reused directly. Preliminary results (not shown) suggest these autoregressive features can substantially improve downstream performance on both the sleep staging and pathology detection tasks.

Although the proposed tasks proved successful, many other pretext tasks could have been designed based on an intuitive understanding of EEG. For instance, a transformation discrimination task was applied to ECG data in [16] and could be adapted for EEG. Similarly, nonlinear ICA-derived frameworks such as time contrastive learning [77] and generalized contrastive learning [31] could be used to explicitly leverage nonstationnarity or other structure present in EEG signals.

4.4 Limitations

We identify three principal limitations to this work: fixed training hyperparameters across data regimes, restricted architecture search, and difference between reported results and state of the art.

Deep neural networks can easily overfit the training data when few examples are available, which can negatively impact generalization. Typical ways of addressing overfitting include increasing the strength of regularization such as dropout and weight decay and performing early stopping. Given the computational requirements of training neural networks on large EEG datasets, we fixed the training hyperparameters of the fully supervised models (i.e., learning rate, batch size, dropout, weight decay) and reused the same values across all data regimes. As a result, the fully supervised models typically stopped learning after only a few epochs, although they might have been able to train longer with different hyperparameters. We tested the impact of various training hyperparameter settings on a subset of the models and saw that even though training can be slightly improved by changing hyperparameters, this effect is not strong enough to change any of our conclusions (results not shown).

Similarly, hyperparameter search was limited to pretext task hyperparameters in our experiments. However, architecture hyperparameters (e.g., number of convolutional channels, dimensionality of the embedding, number of layers, etc.) can also play a critical role in achieving high performance using SSL [78]. Sticking to a single fixed architecture for all models and data regimes means that these improvements - which could help bridge (or widen) the gap between SSL methods and the various baselines - were not taken into account in this work.

Finally, the goal of this work being to introduce self-supervision as a representation learning paradigm for EEG, we did not focus on matching state-of-the-art performance from earlier work on sleep staging and pathology detection. To the best of our knowledge, we are the first to present sleep staging results on PC18, and therefore there is no prior art to which our results can be fairly compared. Nonetheless, downstream performance would most likely improve significantly by aggregating temporal windows as was shown on other datasets [42, 76]. Similarly, on TUHab, we chose to reuse simpler approaches from [48] instead of the best performing models. Moreover, (1) we did not use the cropped decoding approach presented in [53], (2) we used a simpler normalization methodology (z-score instead of exponential moving average normalization) and (3) our train/validation data split was different. Together, these differences explain the small drop in performance between these state-of-the-art methods and the results reported here.

5 Conclusion

In this work, we introduced SSL approaches to learn representations on EEG data and showed that they could compete with and sometimes outperform traditional supervised approaches on two large clinical tasks. Importantly, the features learned by SSL displayed a clear structure in which different physiological quantities were jointly encoded. This validates the potential of self-supervision to capture important physiological information even in the absence of labeled data. Future work will have to demonstrate whether SSL can also be used successfully with other kinds of EEG recordings and tasks, such as regression. Ultimately, developing a better understanding of how a pretext task can be designed to target a specific kind of EEG structure will be critical in establishing self-supervision as a key component of any EEG analysis pipeline.

Acknowledgements

This work was supported by Mitacs (project number IT14765) and InteraXon Inc. (graduate funding support) for HB, by the ANR BrAIN grant for AG and DAE, and by a Fellowship from CIFAR and from the DATAIA convergence institute as part of the “Programme d’Investissement d’Avenir” (ANR-17-CONV-0003) operated by Inria for AH.

References

[1] Mohammad M Ghassemi, Benjamin E Moody, Li-Wei H Lehman, Christopher Song, Qiao Li, Haoqi Sun, Roger G Mark, M Brandon Westover, and Gari D Clifford. You snooze, you win: the physionet/computing in cardiology challenge 2018. In *2018 Computing in Cardiology Conference (CinC)*, volume 45, pages 1–4. IEEE, 2018.

[2] U Rajendra Acharya, S Vinitha Sree, G Swapna, Roshan Joy Martis, and Jasjit S Suri. Automated EEG analysis of epilepsy: a review. *Knowledge-Based Systems*, 45:147–165, 2013.

[3] Fabien Lotte, Laurent Bougrain, Andrzej Cichocki, Maureen Clerc, Marco Congedo, Alain Rakotomamonjy, and Florian Yger. A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update. *Journal of Neural Engineering*, 15(3):031005, 2018.

[4] Vojkan Mihajlović, Bernard Grundlehner, Ruud Vullers, and Julien Penders. Wearable, wireless EEG solutions in daily life applications: What are we missing? *IEEE Journal of Biomedical and Health Informatics*, 19(1):6–21, 2015.

[5] Alexander J Casson. Wearable EEG and beyond. *Biomedical Engineering Letters*, 9(1):53–71, 2019.

[6] Raman K Malhotra and Alon Y Avidan. Sleep stages and scoring technique. *Atlas of Sleep Medicine*, pages 77–99, 2013.

[7] Magdy Younes, Jill Raneri, and Patrick Hanly. Staging sleep in polysomnograms: Analysis of inter-scorer variability. *Journal of Clinical Sleep Medicine*, 12(06):885–894, 2016.

[8] Denis A Engemann, Federico Raimondo, Jean-Rémi King, Benjamin Rohaut, Gilles Louppe, Frédéric Faugeras, Jitka Annen, Helena Cassol, Olivia Gosseries, Diego Fernandez-Slezak, Steven Laureys, Lionel Naccache, Stanislas Dehaene, and Jacobo D Sitt. Robust EEG-based cross-site and cross-protocol classification of states of consciousness. *Brain*, 141(11):awy251, 2018.

[9] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *arXiv preprint arXiv:1902.06162*, 2019.

[10] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pages 69–84. Springer, 2016.

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

[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[14] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4918–4927, 2019.

[15] Ye Yuan, Guangxu Xun, Qiuling Suo, Kebin Jia, and Aidong Zhang. Wave2Vec: Learning deep representations for biosignals. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 1159–1164, Nov 2017.

[16] Pritam Sarkar and Ali Etemad. Self-supervised ECG representation learning for emotion recognition. *arXiv preprint arXiv:2002.03898*, 2020.

[17] 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, 2019.

[18] Denis A Engemann, Oleh Kozynets, David Sabbagh, Guillaume Lemaître, Gael Varoquaux, Franziskus Liem, and Alexandre Gramfort. Combining magnetoencephalography with magnetic resonance imaging enhances learning of surrogate-biomarkers. *eLife*, 9:e54055, 2020.

[19] Iyad Obeid and Joseph Picone. The temple university hospital EEG data corpus. *Frontiers in Neuroscience*, 10:196, 2016.

[20] Clare Bycroft, Colin Freeman, Desislava Petkova, Gavin Band, Lloyd T Elliott, Kevin Sharp, Allan Motyer, Damjan Vukcevic, Olivier Delaneau, Jared O'Connell, et al. The UK Biobank resource with deep phenotyping and genomic data. *Nature*, 562(7726):203–209, 2018.

[21] Meredith A Shafto, Lorraine K Tyler, Marie Dixon, Jason R Taylor, James B Rowe, Rhodri Cusack, Andrew J Calder, William D Marslen-Wilson, John Duncan, Tim Dalgleish, et al. The Cambridge Centre for Ageing and Neuroscience (Cam-CAN) study protocol: a cross-sectional, lifespan, multidisciplinary examination of healthy cognitive ageing. *BMC neurology*, 14(1):204, 2014.

[22] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, pages 1422–1430, 2015.

[23] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, pages 527–544. Springer, 2016.

[24] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Association for Computational Linguistics Annual Meeting*, pages 384–394. Association for Computational Linguistics, 2010.

[25] Neha Nayak, Gabor Angeli, and Christopher D Manning. Evaluating word embeddings using a representative suite of practical tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 19–23, 2016.

[26] Olivier J Hénaff, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

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

[28] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[29] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[30] Aapo Hyvärinen and Hiroshi Morioka. Nonlinear ICA of temporally dependent stationary sources. In *AISTATS*, 2017.

[31] Aapo Hyvärinen, Hiroaki Sasaki, and Richard E Turner. Nonlinear ICA using auxiliary variables and generalized contrastive learning. In *AISTATS*, 2019.

[32] Scott Makeig, Tzyy-Ping Jung, Anthony J Bell, Dara Ghahremani, and Terrence J Sejnowski. Blind separation of auditory event-related brain responses into independent components. *Proceedings of the National Academy of Sciences*, 94(20):10979–10984, 1997.

[33] Tzyy-Ping Jung, Colin Humphries, Te-Won Lee, Scott Makeig, Martin J McKeown, Vicente Iragui, and Terrence J Sejnowski. Extended ICA removes artifacts from electroencephalographic recordings. In *Advances in neural information processing systems*, pages 894–900, 1998.

[34] Lucas C Parra, Clay D Spence, Adam D Gerson, and Paul Sajda. Recipes for the linear analysis of EEG. *Neuroimage*, 28(2):326–341, 2005.

[35] Pierre Ablin, Jean-François Cardoso, and Alexandre Gramfort. Faster independent component analysis by preconditioning with Hessian approximations. *IEEE Transactions on Signal Processing*, 66(15):4040–4049, 2018.

[36] Suzanna Becker. Learning to categorize objects using temporal coherence. In *Advances in neural information processing systems*, pages 361–368, 1993.

[37] Laurenz Wiskott and Terrence J Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002.

[38] Bruce M Altevogt and Harvey R Colten. *Sleep disorders and sleep deprivation: an unmet public health problem.* National Academies Press, 2006.

[39] 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.

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

[41] Christina Jayne Bathgate and Jack D Edinger. Diagnostic criteria and assessment of sleep disorders. In *Handbook of Sleep Disorders in Medical Conditions*, pages 3–25. Elsevier, 2019.

[42] Stanislas Chambon, Mathieu N Galtier, Pierrick J Arnal, Gilles Wainrib, and Alexandre Gramfort. A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(4):758–769, 2018.

[43] Shayan Motamedi-Fakhr, Mohamed Moshrefi-Torbati, Martyn Hill, Catherine M. Hill, and Paul R. White. Signal processing techniques applied to human sleep EEG signals—a review. *Biomedical Signal Processing and Control*, 10:21 – 33, 2014.

[44] SJM Smith. EEG in the diagnosis, classification, and management of patients with epilepsy. *Journal of Neurology, Neurosurgery & Psychiatry*, 76(suppl 2):ii2–ii7, 2005.

[45] Christina Micanovic and Suvankar Pal. The diagnostic utility of EEG in early-onset dementia: a systematic review of the literature with narrative analysis. *Journal of Neural Transmission*, 121(1):59–69, 2014.

[46] S Lopez, G Suarez, D Jungreis, I Obeid, and J Picone. Automated identification of abnormal adult EEGs. In *2015 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pages 1–5. IEEE, 2015.

[47] Robin Tibor Schirrmeister, Lukas Gemein, Katharina Eggensperger, Frank Hutter, and Tonio Ball. Deep learning with convolutional neural networks for decoding and visualization of EEG pathology. *arXiv preprint arXiv:1708.08012*, 2017.

[48] Lukas AW Gemein, Robin T Schirrmeister, Patryk Chrabąszcz, Daniel Wilson, Joschka Boedecker, Andreas Schulze-Bonhage, Frank Hutter, and Tonio Ball. Machine-learning-based diagnostics of EEG pathology. *Neuroimage*, page 117021, 2020.

[49] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[50] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015.

[52] Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

[53] Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. Deep learning with convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping*, aug 2017.

[54] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A Engemann, Daniel Strohmeier, Christian Brodbeck, Lauri Parkkonen, and Matti S Hämäläinen. MNE software for processing MEG and EEG data. *Neuroimage*, 86:446–460, 2014.

[55] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS*, 2017.

[56] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. Classification of covariance matrices using a Riemannian-based kernel for BCI applications. *Neurocomputing*, 112:172 – 178, 2013. Advances in artificial neural networks, machine learning, and computational intelligence.

[57] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[58] Richard B Berry, Rita Brooks, Charlene E Gamaldo, Susan M Harding, Carole L Marcus, Bradley V Vaughn, et al. The AASM manual for the scoring of sleep and associated events. *Rules, Terminology and Technical Specifications, American Academy of Sleep Medicine*, 176, 2012.

[59] Khald Aboalayon, Miad Faezipour, Wafaa Almuhammadi, and Saeid Moslehpour. Sleep stage classification using EEG signal analysis: a comprehensive survey and new investigation. *Entropy*, 18(9):272, 2016.

[60] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[61] James Pardey, Stephen Roberts, Lionel Tarassenko, and John Stradling. A new approach to the analysis of the human sleep/wakefulness continuum. *Journal of sleep research*, 5(4):201–210, 1996.

[62] Beth A Lopour, Savas Tasoglu, Heidi E Kirsch, James W Sleigh, and Andrew J Szeri. A continuous mapping of sleep states through association of EEG with a mesoscale cortical model. *Journal of computational neuroscience*, 30(2):471–487, 2011.

[63] Bryce A Mander, Joseph R Winer, and Matthew P Walker. Sleep and human aging. *Neuron*, 94(1):19–36, 2017.

[64] SM Purcell, DS Manoach, C Demanuele, BE Cade, S Mariani, R Cox, G Panagiotaropoulou, R Saxena, JQ Pan, JW Smoller, et al. Characterizing sleep spindles in 11,630 individuals from the national sleep research resource. *Nature Communications*, 8:15930, 2017.

[65] Geoffrey F Woodman. A brief introduction to the use of event-related potentials in studies of perception and attention. *Attention, Perception, & Psychophysics*, 72(8):2031–2046, 2010.

[66] Reza Abiri, Soheil Borhani, Eric W Sellers, Yang Jiang, and Xiaopeng Zhao. A comprehensive review of EEG-based brain–computer interface paradigms. *Journal of Neural Engineering*, 16(1):011001, 2019.

[67] Hubert Banville, Graeme Moffat, Isabela Albuquerque, Denis-Alexander Engemann, Aapo Hyvärinen, and Alexandre Gramfort. Self-supervised representation learning from electroencephalography signals. In *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2019.

[68] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning, 2020.

[69] Alfred L Loomis, E Newton Harvey, and Garret A Hobart. Cerebral states during sleep, as studied by human brain potentials. *Journal of experimental psychology*, 21(2):127, 1937.

[70] Anthony Kales, Allan Rechtschaffen, Los Angeles. Brain Information Service University of California, and NINDB Neurological Information Network (U.S.). *A Manual of Standardized Terminology, Techniques and Scoring System for Sleep Stages of Human Subjects: Allan Rechtschaffen and Anthony Kales, Editors*. NIH publication. U. S. National Institute of Neurological Diseases and Blindness, Neurological Information Network, 1968.

[71] Doris Moser, Peter Anderer, Georg Gruber, Silvia Parapatics, Erna Loretz, Marion Boeck, Gerhard Kloesch, Esther Heller, Andrea Schmidt, Heidi Danker-Hopfe, et al. Sleep classification according to AASM and Rechtschaffen & Kales: effects on sleep scoring parameters. *Sleep*, 32(2):139–149, 2009.

[72] Hartmut Schulz. Rethinking sleep analysis comment on the aasm manual for the scoring of sleep and associated events. *Journal of Clinical Sleep Medicine*, 4(02):99–103, 2008.

[73] Alison M Pack. Epilepsy overview and revised classification of seizures and epilepsies. *CONTINUUM: Lifelong Learning in Neurology*, 25(2):306–321, 2019.

[74] Mary Jane England, Catharyn T Liverman, Andrea M Schultz, and Larisa M Strawbridge. Epilepsy across the spectrum: Promoting health and understanding.: A summary of the institute of medicine report. *Epilepsy & Behavior*, 25(2):266–276, 2012.

[75] Glenn D Walters. Dementia: Continuum or distinct entity? *Psychology and aging*, 25(3):534, 2010.

[76] 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.

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

[78] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1920–1929, 2019.

A UMAP on RP-learned features

We show the results of the experiment of Section 3.2 for features learned with the RP pretext task in Figures 8 and 9.

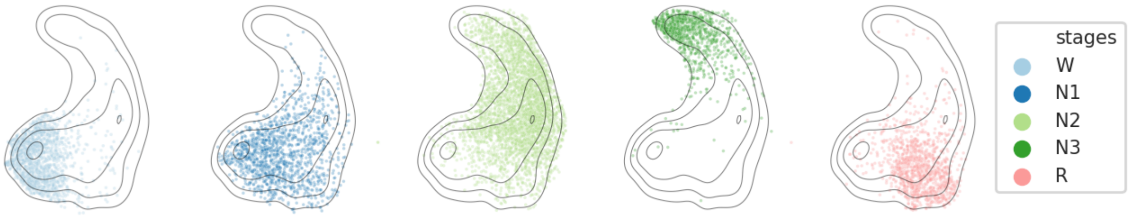

Figure 8: UMAP visualization of RP-learned features on the entire PC18 dataset. See Fig. 4 for a complete description.

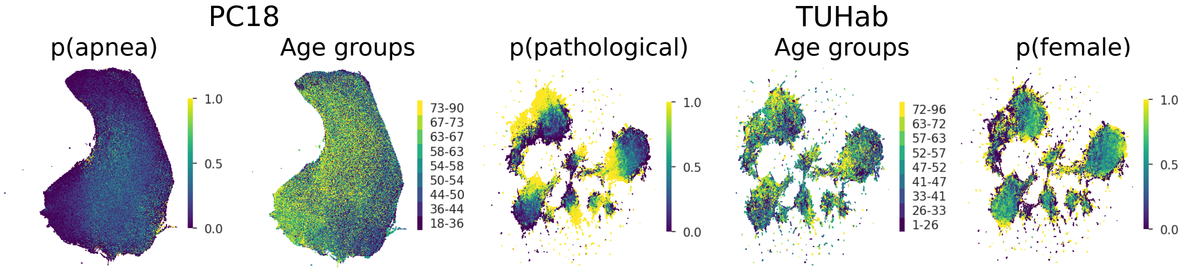

Figure 9: Structure learned by the embedders trained on the RP task. See Fig. 5 for a complete description.

B Hyperparameter search procedure

The hyperparameter search in Section 3.3 was carried out using the following steps. First, embedders h_Θ were independently trained on the RP, TS and CPC tasks. The parameters of h_Θ were then frozen, and the different h_Θ were used as feature extractors to obtain sets of 100-dimensional feature vectors from the original input data. Finally, we trained linear logistic

	PC18		**TUHab**	
	τ_{pos} and τ_{neg} (min)	τ_{neg} (min)	τ_{pos} and τ_{neg} (min)	τ_{neg} (min)
RP	0.5, 1, 2, 5, 15, 30, 60, 120	0, 0.5, 1, 2, 5, 15, 30, 60, 120	6 s, 12 s, 0.5, 1, 2, 5, 10	0 s, 6 s, 12 s, 0.5, 1, 2, 5, 10
TS	1, 2, 5, 15, 30, 60, 120	2, 5, 15, 30, 60, 120	12 s, 0.5, 1, 2, 5, 10	12 s, 0.5, 1, 2, 5, 10
	# predicted windows	Negative sampling	# predicted windows	Negative sampling
CPC	2, 4, 8, 12, 16	same recording, across recordings	2, 4, 8, 12, 16	same recording, across recordings

Table 4: SSL pretext task hyperparameter values considered in Experiment 2.

regression classifiers to perform the downstream tasks given the extracted features. We further varied the principal pretext task hyperparameters to understand their impact on both pretext and downstream task performance (see Table 4). In both cases, we compared the balanced accuracy on the validation set. For RP and TS, we focused our attention on τ_{pos} and τ_{neg}, which are used to control the size of the positive and negative contexts when sampling pairs or triplets of windows. As a first step, the values of τ_{pos} and τ_{neg} were varied jointly, i.e., $\tau_{pos} = \tau_{neg}$, to avoid sampling "confusing" pairs or triplets of windows which could come from either the positive or negative classes. The best value was then used to set τ_{pos}, and a sweep over different τ_{neg} values was carried out. In a second step, we fixed τ_{neg} such that it encompassed all recordings, i.e., negative windows were uniformly sampled from any recording in the dataset instead of being limited to the recording which contains the anchor window. We then again varied τ_{pos} with this second negative sampling strategy. For CPC, we studied the impact of the number of predicted windows ("# steps") [11] and, as for RP and TS, the type of negative sampling ("same-recording" vs. "across-recordings"). Again, we first varied the number of predicted windows and reused the best value to compare negative sampling strategies.