Self-Supervised Learning for Electroencephalograms

Théo Moutakanni
École CentraleSupélec
theo.moutakanni@supelec.fr

Clément Bonnet
École CentraleSupélec
clement.bonnet16@gmail.com

Abstract

Supervised learning leads to better and better results thanks to the advances made using Deep Learning. The main problem with supervision is the need to have more and more labeled data to increase performance. In particular, medical data is subject to the lack of labels. It requires experienced personnel to obtain quality labels and the time needed to obtain them to the same order of magnitude as for conventional tasks is prohibitive. New methods with little or no supervision then emerged, particularly in computer vision. We will focus on so-called self-supervised learning methods, specifically on contrastive methods.

TO BE CONTINUED ...

1 Introduction

Given enough data and labels, supervised learning can solve a given task really well. However, good performance may require huge amounts of labels which are often collected manually, resulting in a costly and rather not scalable process. The internet gathers way more unlabeled data than any human-labeled dataset. Therefore, one may find it wasteful not to use these unlabeled data when training algorithms. Unsupervised learning, whose goal is to learn about the data generation process while only using unlabeled data, is rather difficult and not as good as supervised learning on a given final task.

Self-supervised learning (SSL) enables the supervision to come from the data itself. This way, one can make use of the much larger amount of unlabeled data that is generated all the time. The self-supervised task, sometimes called the pretext task, sets the learning objective to be a supervised loss function. This invented task, whose final performance is rather ignored, is expected to learn an intermediate representation that carries the structural meaning of the data generation process, which in turn, is used to learn downstream tasks from few examples. This way, one uses unlabeled data in a supervised way to learn a representation to be provided to the downstream task as an inductive bias. Self-supervised learning cares about producing good features for other tasks.

...

Research project at École CentraleSupélec.

2 Background

2.1 Electroencephalograms

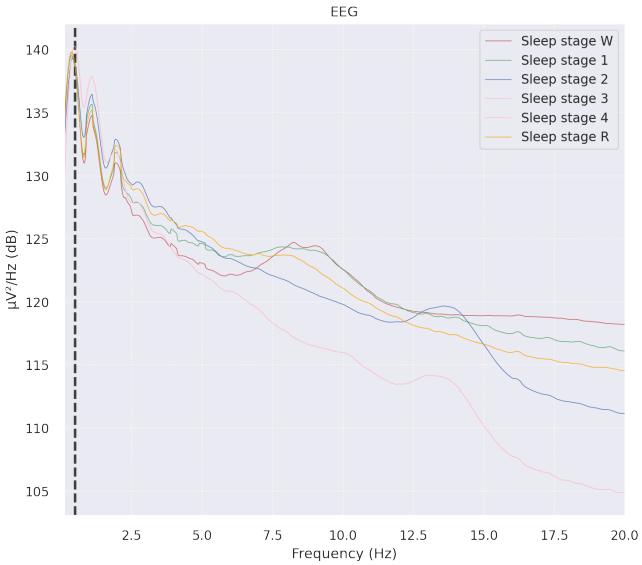


Figure 1: Mean PSD by sleep stages.

2.2 Self-Supervised Learning

Utilise le biais intrinsèque aux données différentes méthodes - contrastive (ce qu'on fait, negative vs positive) - clustering (faire des paquets) - autres (truc fancy comme BYOL...)

2.3 Application to Sleep Staging

3 Related Work

3.1 Self-Supervised Learning

Many ideas have been proposed so far for self-supervised representation learning of images, videos and times series. Thanks to useful resources including Lilian Weng's blog [1], we provide below an overview of these methods in each of these machine learning fields.

3.1.1 Images

As we mentioned above, the internet is overflowing with unlabeled images. A common workflow for self-supervised representation learning of images is to train a model on one or several pretext tasks, and then use an intermediate layer of the model as a basis for a supervised learning architecture eventually trained on ImageNet [2] or any labeled dataset.

Some inductive biases used in SSL for images are that the spatial context of objects should be understood by a model. One way to achieve this is to frame a pretext task to recover the relative positions of random patches extracted from the same image. In Doersch et al. [3], the authors described the following SSL procedure. A first patch is randomly selected from an image. Then, a second patch is randomly chosen among its 8 neighbors in a 3x3 grid. The model is trained to predict at which of the 8 neighboring areas the second patch was located. This way, the model is forced to understand contextual information of objects in images.

3.1.2 Videos

When dealing with videos as sequences of frames, a common workflow for self-supervised learning of images is to train a model on a pretext task with unlabeled videos and use one of its intermediate features layers to train a simple model on a final task such as object tracking or video classification. Here, the idea is to introduce inductive biases such as to take physical principles into account. For instance, nearby frames must be correlated in a way that does not allow objects to move too fast, or that gravity tends to make things fall.

One way to use SSL to learn a good representation of video frames is to expect a model to be able to correctly order frames of the same video. Naturally, albeit being unlabeled, video frames are positioned in chronological order. Therefore, one may learn a visual representation from the raw spatiotemporal signals in videos. In Misra et al. [4], the authors formulate the representation learning task as an unsupervised sequential verification task in which a model has to determine whether a sequence of frames from the same video is in the correct temporal order. This way, the model is forced to reason and track small motions of objects across frames to succeed in such a pretext task.

3.1.3 Time Series

Self-supervised learning may be used in the context of multivariate times series. A successful approach used in Banville et al. [5] is to predict whether time windows are sampled from the same temporal context or not. From electroencephalogry signals, the authors learn a representation using a temporal contrastive task that they describe as follows. From the multivariate times series, they produce labels by sampling pairs of time windows from the same long time series. The main assumption used by the authors is that an appropriate representation of the data should evolve slowly over time suggesting that time windows close in time should share the same label. They come upon two self-supervised tasks based on this assumption.

The first task used in [5] and [6] is called Relative Positioning. The model is trained to determine whether two sampled time windows are close or not. The second idea of the authors is to train the model with another task called Temporal Shuffling. This time, the task it to predict if three sampled windows are in the correct chronological order or if they have been shuffled. This way, the model learns a good representation of the multivariate time series.

3.2 Sleep staging detection

Sleep stages are usually determined by a sleep expert by analyzing polysomnography records (PSG). The expert analyzes 30-seconds windows and classifies them using a normalized protocol to get the sleep stages. Each window gets a label between N1, N2, N3 and REM, the 4 sleep stages. This process is time expensive, but is necessary as a first step to diagnose sleep disorders.

There exists a lot of published processes to automatize this procedure. Newer ones perform really well on the datasets they are trained with [7] [8] [9] [10]. They all use Deep Learning, which needs a lot of training labeled data. Furthermore, as shown by Guillot et al. [10], they are not robust enough to unseen PSG montage or new cohort of patients. In fact, there are a multitude of PSG devices available on the market and each hospital may use several ones. Methods trained on datasets with unseen diseases also have bad results on patients with multiple comorbidities.

3.3 Domain Adaptation

The issue automatic sleep staging algorithms have to overcome to be able to be used in real life is called Domain Adaptation. The goal is to reduce the domain bias between datasets such that an algorithm trained on one fixed dataset can be applied on new ones from hospitals. There are three major domain bias that can be found. Different EEG montage or devices (hardware), different patient cohorts (with or without comorbidities, age, sex, job category etc.), and patient-specific differences (position, brain connectivity, sleep patterns etc.). While some forms can be important (patient-specific bias), domain bias should be reduced and sleep staging algorithms should be robuster.

Several supervised algorithm exists, some specific to sleep staging [11], and others more general to Deep Learning [12].

A more straight forward method is to first train the network on a big fixed dataset, and then fine-tune it on a labeled subset of the new unseen dataset which is order of magnitude smaller than the number of necessary samples if the network was trained from scratch. This method is effective because the first layers of the network act as feature extractors and can be shared among tasks and datasets [13] [14]. By doing so, we can increase the sleep staging results on newer datasets without access to a lot of labeled data.

According to Guillot et al. [10], direct transfer (no fine-tuning) already leads to good results with the RobustSleepNet architecture (90% to more than 100% of the original F1 if the network was trained on the unseen dataset directly). With only 10 to 40 labeled records, the fine-tuned pretrained network surpasses a network trained directly on the datasets used for evaluation.

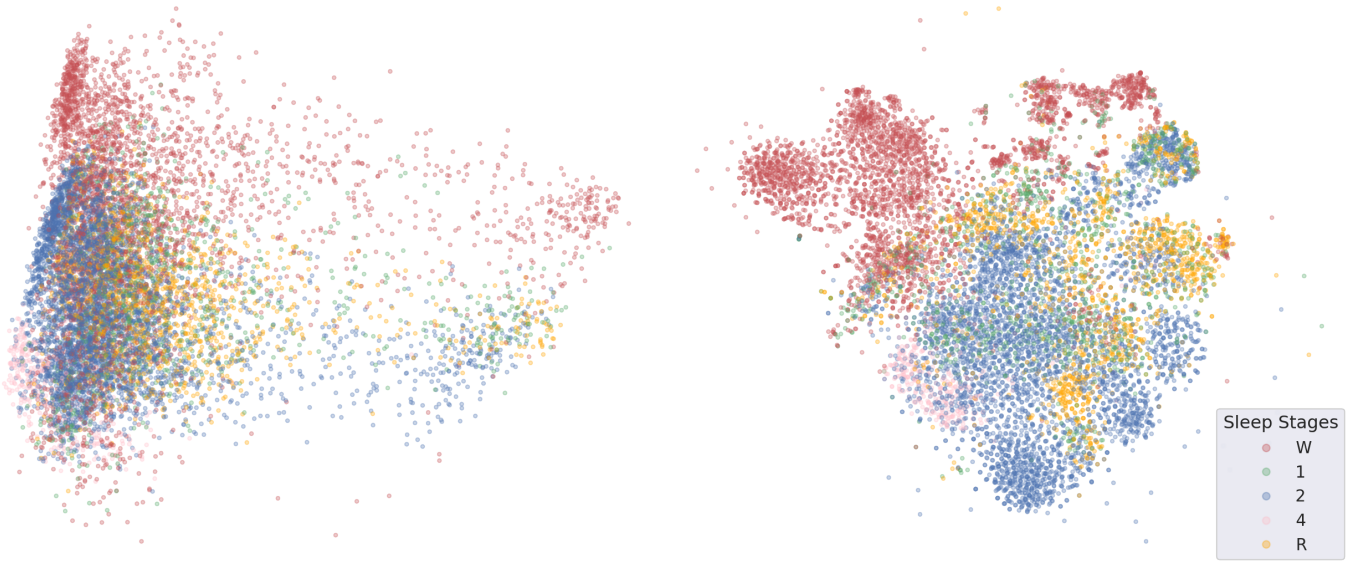
4 Experiments

4.1 Dataset

Dataset [15]

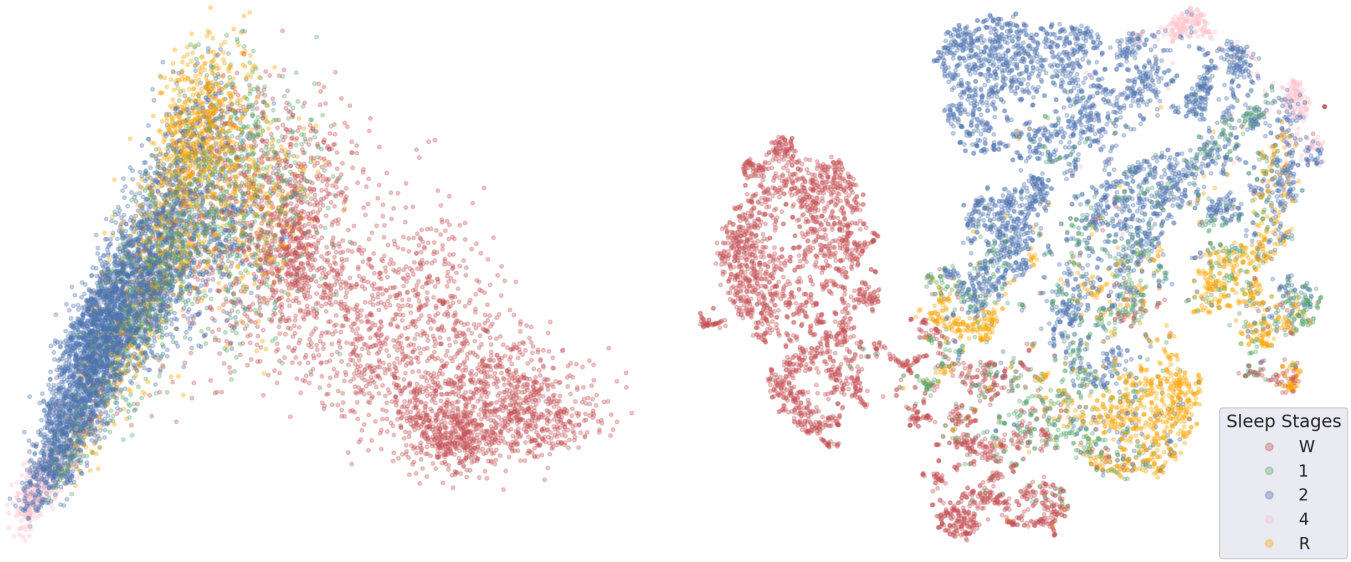
4.2 Proof Of Concept

4.2.1 Visualization of the embeddings



(a) PCA (b) TSNE

Figure 2: Visualization of the PSD features according to the sleep stages.



(a) PCA (b) TSNE

Figure 3: Visualization of the SSL embeddings according to the sleep stages.

4.2.2 Benchmark of the embeddings

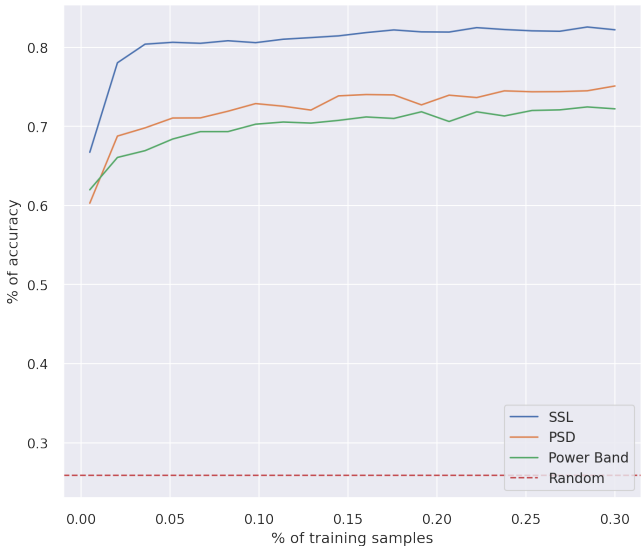


Figure 4: Sleep staging benchmark of 3 unsupervised embedding methods using KNN trained on a fraction of the test data.

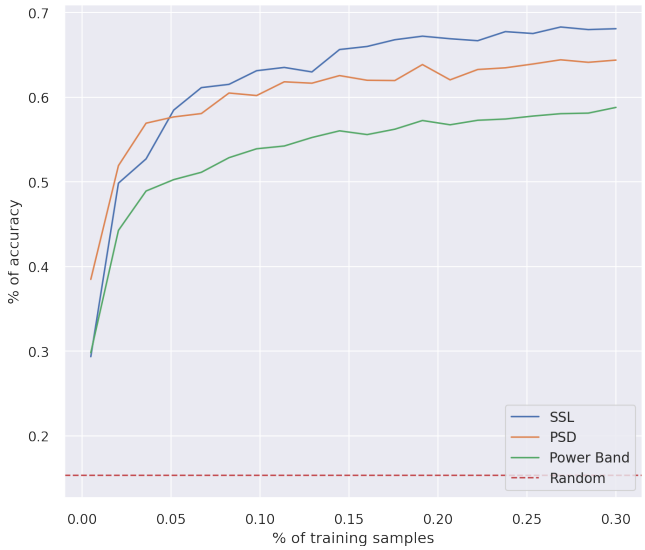


Figure 5: Patient ID detection benchmark of 3 unsupervised embedding methods using KNN trained on a fraction of the test data.

4.3 Fine tuning

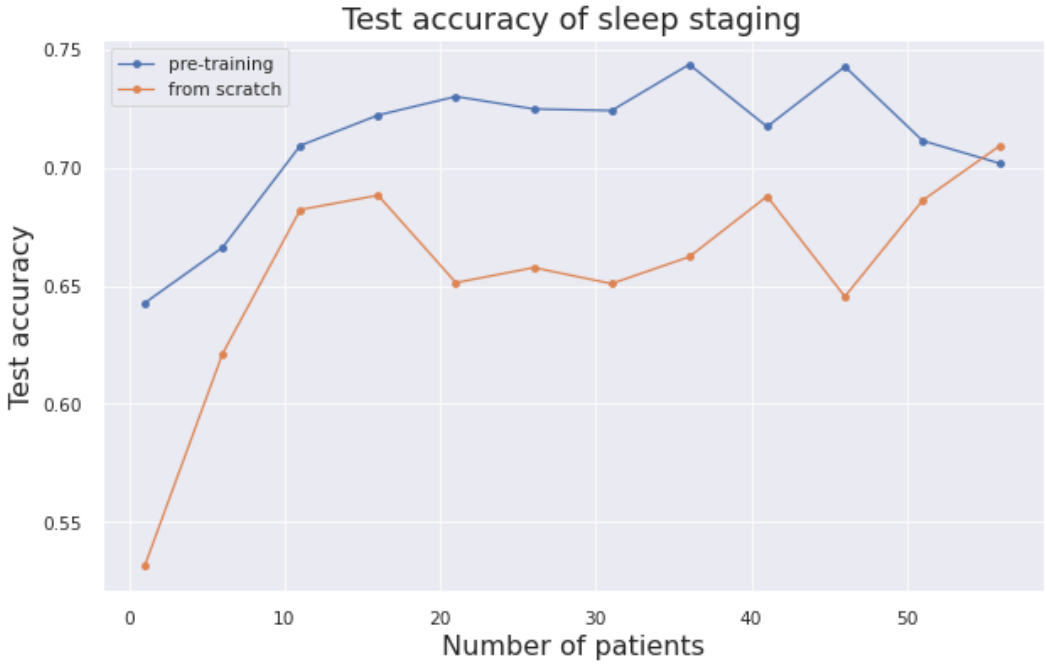
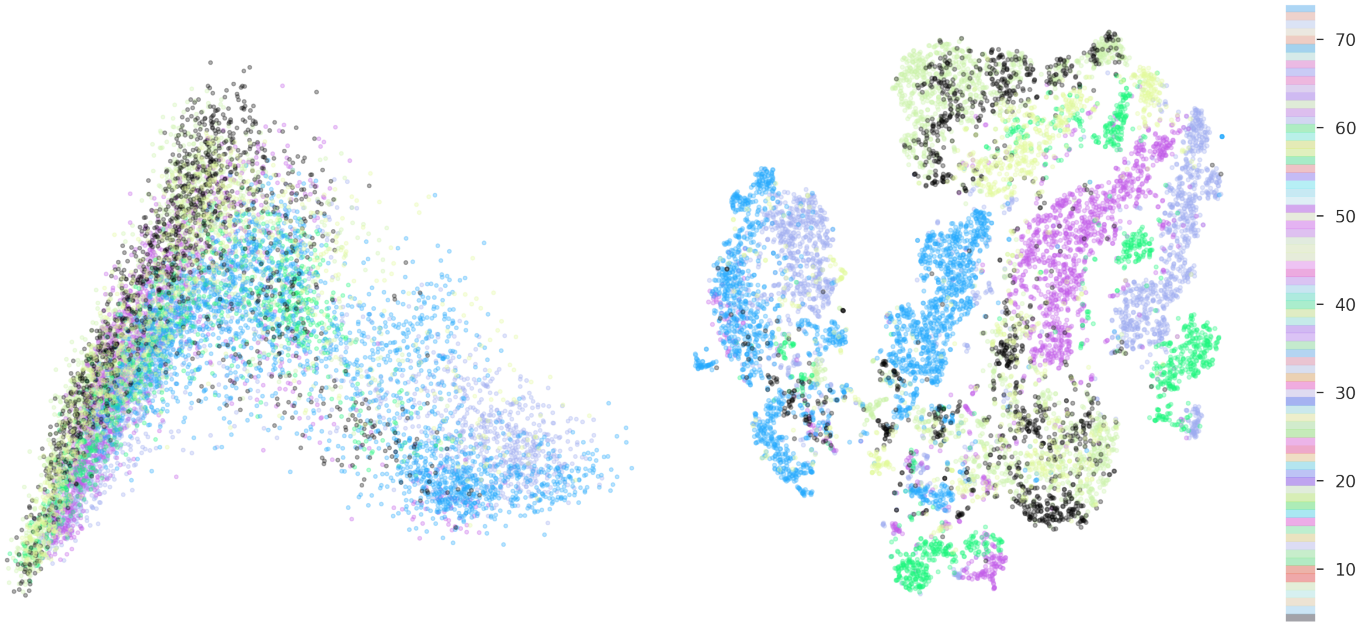


Figure 6: Fine-tuning on the SSL embeddings for sleep staging.

4.4 Analysis of Learned Representation

(a) PCA (b) TSNE

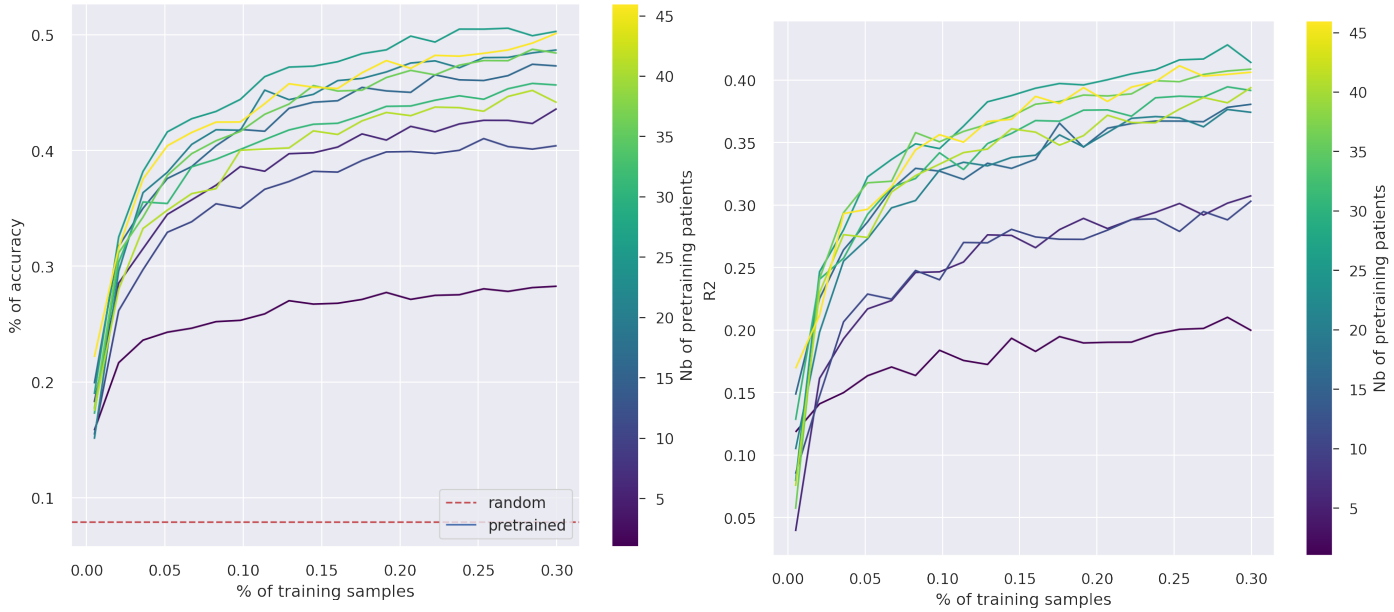
Figure 7: Visualization of the embeddings according to the age of the patient.



(a) PCA (b) TSNE

Figure 8: Visualization of the embeddings according to the patient ID.

4.5 Impact of Pre-Training Set Size



(a) Accuracy when predicting patient ID

(b) R2 when predicting age

Figure 9: Detection of some patient information using KNN trained on a small subset of the test patients.

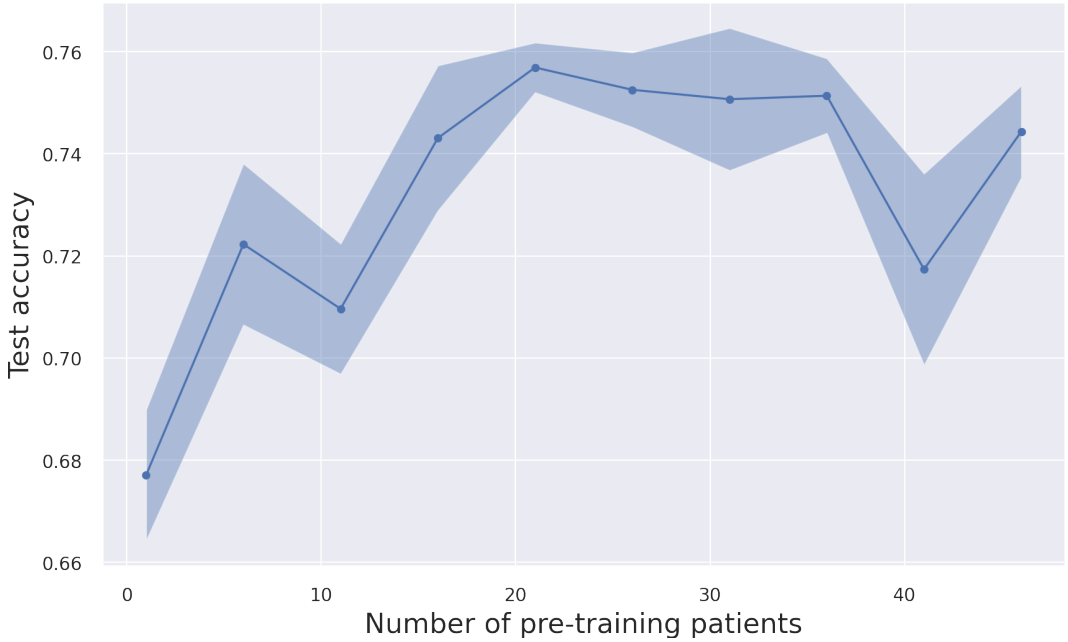


Figure 10: Fine tuning accuracy when changing the number of pre-training patients.

5 Conclusion

Acknowledgments

We would like to thank our supervisors Alexandre Gramfort and Nora Ouzir for their feedback and support. The code for the experiments is available at `https://github.com/TheoMoutakanni/TCC-EEG/`.

References

[1] Lilian Weng. Self-supervised representation learning. *lilianweng.github.io/lil-log*, 2019.

[2] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[3] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction, 2016.

[4] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Shuffle and learn: Unsupervised learning using temporal order verification, 2016.

[5] Hubert Banville, Isabela Albuquerque, Aapo Hyvärinen, Graeme Moffat, Denis-Alexander Engemann, and Alexandre Gramfort. Self-supervised representation learning from electroencephalography signals, 2019.

[6] Hubert Banville, Omar Chehab, Aapo Hyvarinen, Denis Engemann, and Alexandre Gramfort. Uncovering the structure of clinical eeg signals with self-supervised learning. *Journal of Neural Engineering*, 2020.

[7] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos. Seqsleepnet: End-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(3):400–410, 2019.

[8] Jens Stephansen, Alexander Olesen, Mads Olsen, Aditya Ambati, Eileen Leary, Hyatt Moore, Oscar Carrillo, Ling Lin, Fang Han, Han Yan, Yun Sun, Yves Dauvilliers, Sabine Scholz, Lucie Barateau, Birgit Högl, Ambra Stefani, Seung Hong, Tae Kim, Fabio Pizza, and Emmanuel Mignot. Neural network analysis of sleep stages enables efficient diagnosis of narcolepsy. *Nature Communications*, 9, 12 2018.

[9] Mathias Perslev, Michael Hejselbak Jensen, Sune Darkner, Poul Jørgen Jennum, and Christian Igel. U-time: A fully convolutional network for time series segmentation applied to sleep staging, 2019.

[10] Antoine Guillot and Valentin Thorey. Robustsleepnet: Transfer learning for automated sleep staging at scale, 2021.

[11] S. Chambon, M. N. Galtier, and A. Gramfort. Domain adaptation with optimal transport improves eeg sleep stage classifiers. In *2018 International Workshop on Pattern Recognition in Neuroimaging (PRNI)*, pages 1–4, 2018.

[12] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks, 2016.

[13] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[14] Mohsen Ghafoorian, Alireza Mehrtash, Tina Kapur, Nico Karssemeijer, Elena Marchiori, Mehran Pesteie, Charles R. G. Guttmann, Frank-Erik de Leeuw, Clare M. Tempany, Bram van Ginneken, Andriy Fedorov, Purang Abolmaesumi, Bram Platel, and William M. Wells. Transfer learning for domain adaptation in mri: Application in brain lesion segmentation. In Maxime Descoteaux, Lena Maier-Hein, Alfred Franz, Pierre Jannin, D. Louis Collins, and Simon Duchesne, editors, *Medical Image Computing and Computer Assisted Intervention MICCAI 2017*, pages 516–524, Cham, 2017. Springer International Publishing.

[15] Ary L. Goldberger, Luis A. N. 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. *Circulation*, 101(23):e215–e220, 2000.

A Evolution of Learned Embeddings as a Function of Pre-Training Size

A.1 Sleep Staging

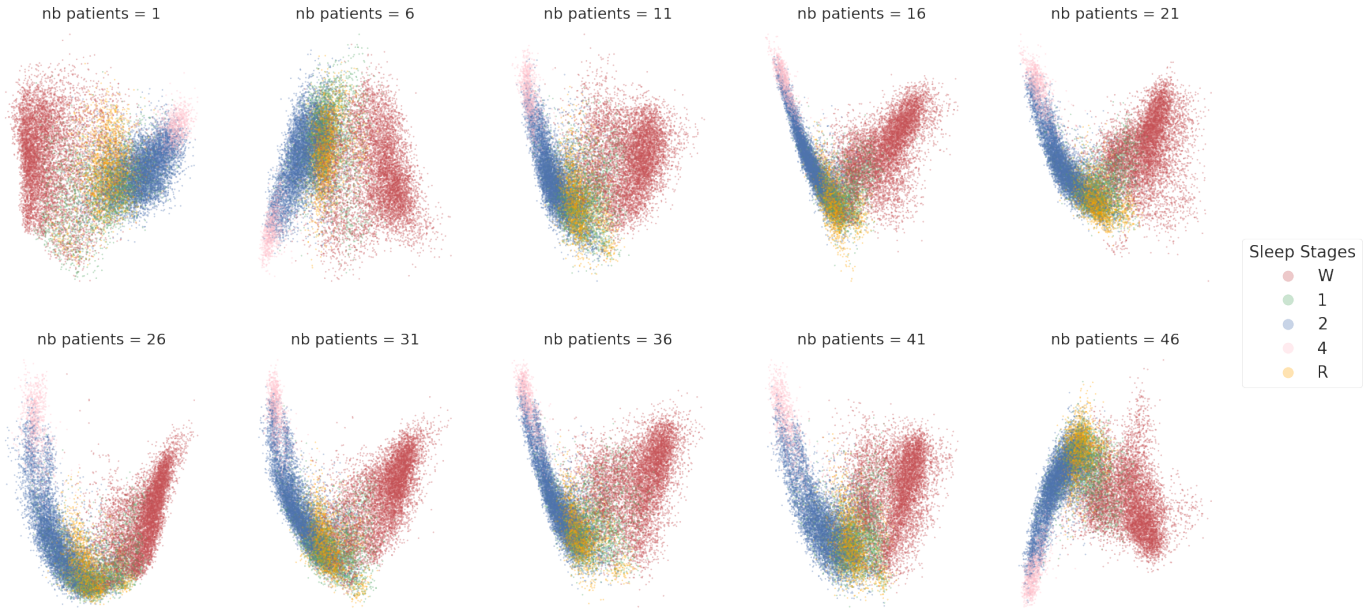


Figure 11: Evolution of PCA according to sleep stages

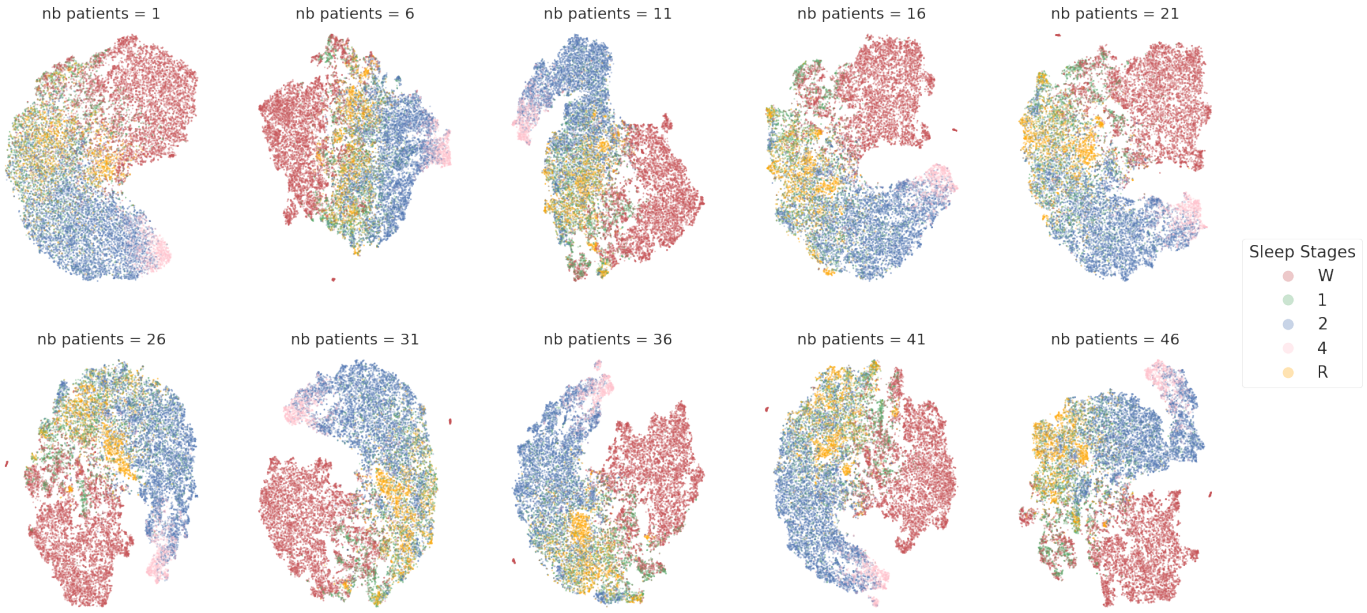


Figure 12: Evolution of T-SNE according to sleep stages

A.2 Patient Age

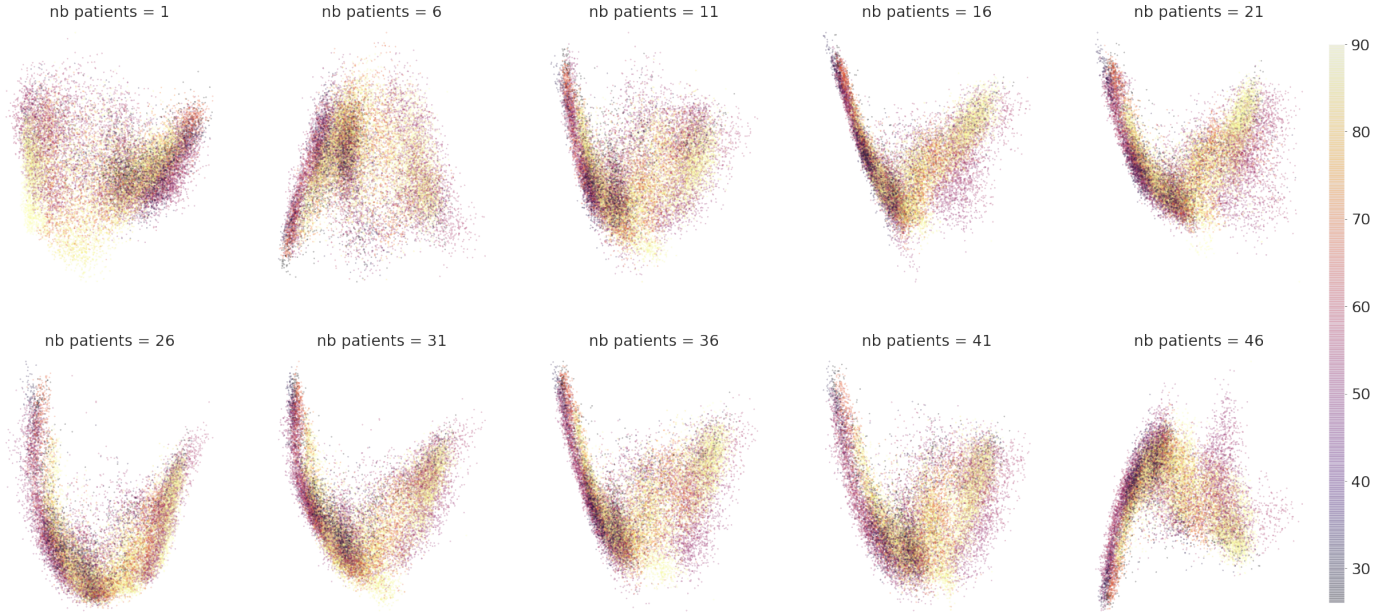


Figure 13: Evolution of PCA according to patient age

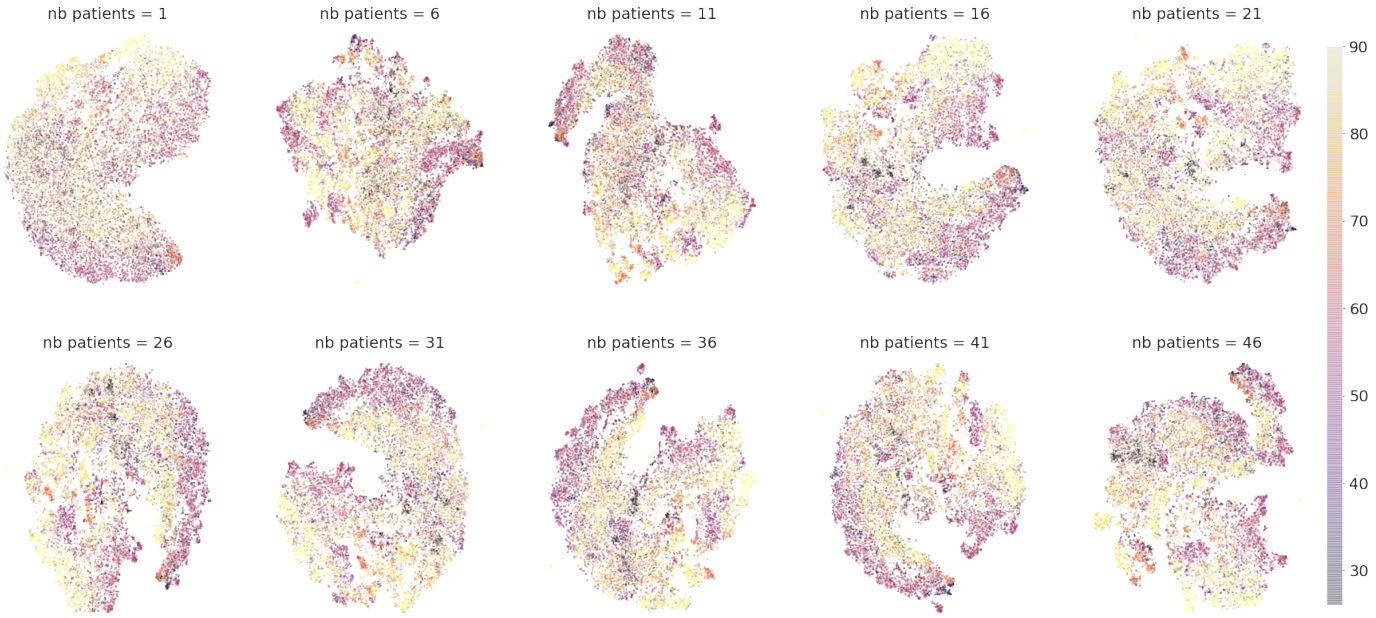


Figure 14: Evolution of T-SNE according to patient age

A.3 Patient ID

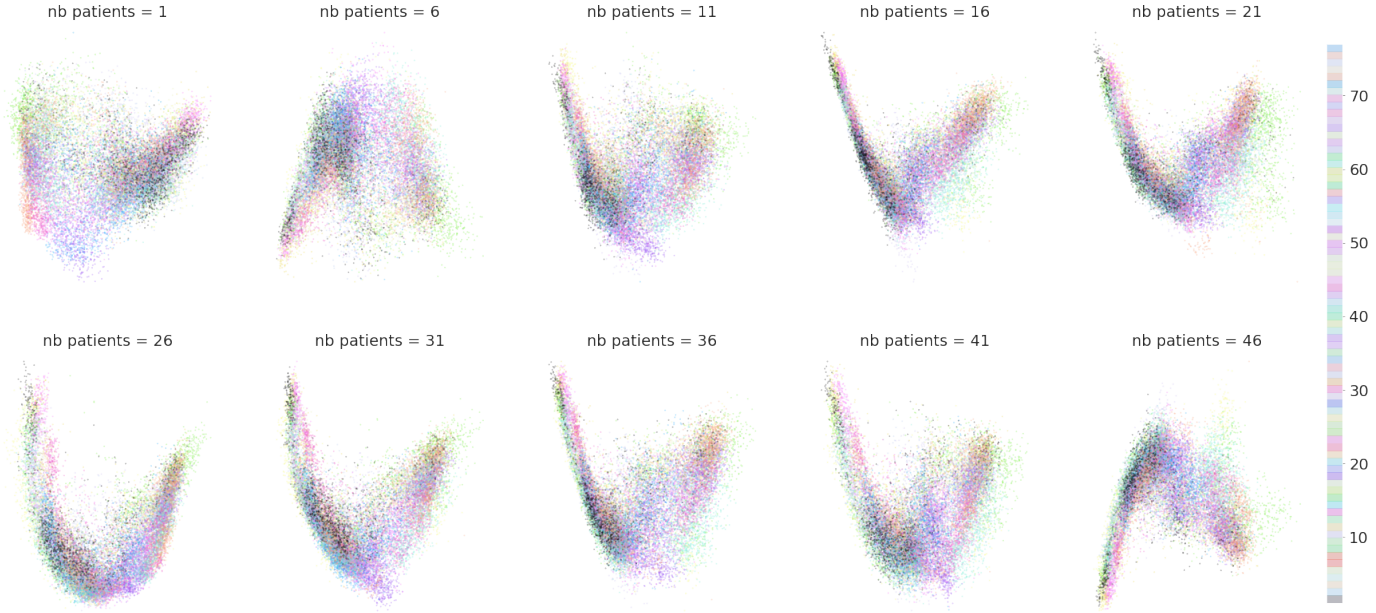


Figure 15: Evolution of PCA according to patient ID

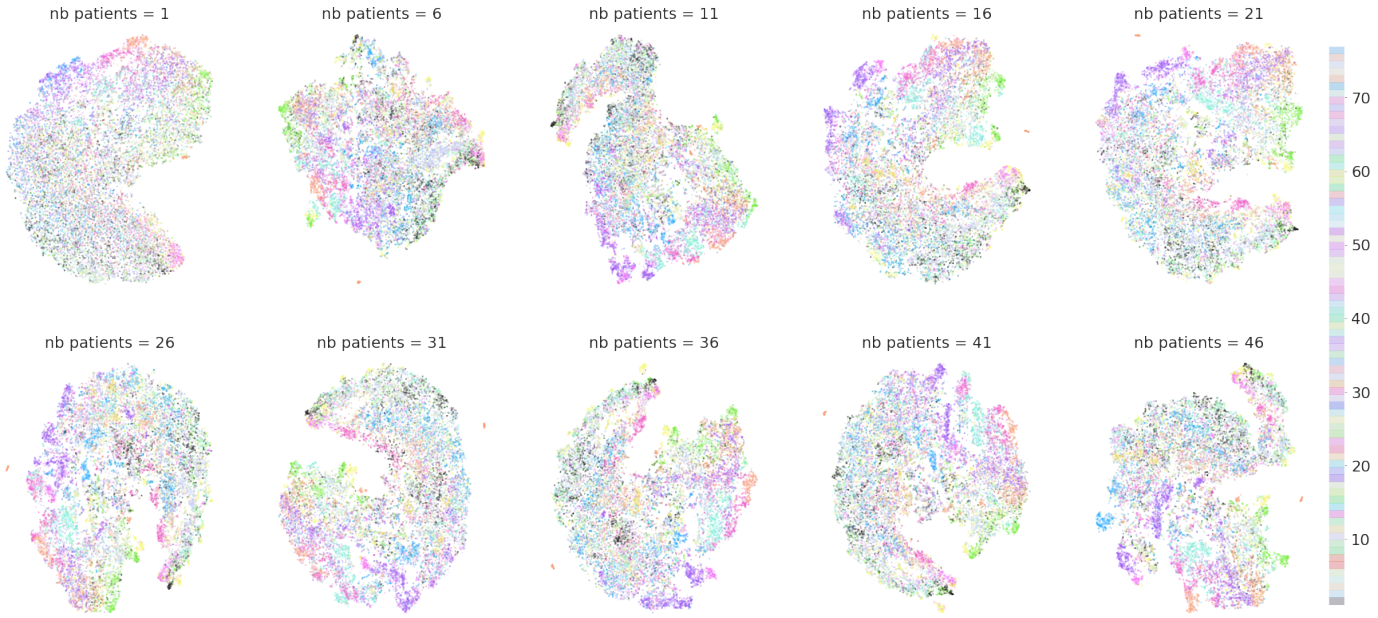


Figure 16: Evolution of T-SNE according to patient ID