Self-Supervision Can Generalize across Domains in Convolutional EEG Models

Etienne Galea[1], Marina Diachenko[1],
Peter Bloem[1], Hilgo Bruining[2–4], and
Klaus Linkenkaer Hansen[1]

[1] Vrije Universiteit, De Boelelaan 1111 1081 HV, Amsterdam, Netherlands
[2] Child and Adolescent Psychiatry and Psychosocial Care, Emma Children's Hospital, Amsterdam UMC, Amsterdam, 1105 AZ, The Netherlands,
[3] N=You Neurodevelopmental Precision Center, Amsterdam Neuroscience, Amsterdam Reproduction and Development, Amsterdam UMC, Amsterdam, 1105 AZ, The Netherlands,
[4] Levvel, Center for Child and Adolescent Psychiatry, Amsterdam, The Netherlands
egalea.11@gmail.com, m.diachenko@vu.nl,
p.bloem@vu.nl, h.bruining@amsterdamumc.nl,
k.linkenkaerhansen@vu.nl

Abstract. Electroencephalographic (EEG) data can be very *domain*-specific: the hardware, subjects and tasks used to acquire the data can all influence the raw signal, obscuring the patterns which persist between settings. Self-supervised learning (SSL) has recently been gaining recognition as a promising means for extracting general features from EEG data. For instance, it has been shown that a simple convolutional neural network can benefit from self-supervised pre-training within one domain. It has also been shown that more complex, transformer-based models can generalize between domains: that is, pre-training on data sampled in one setting can benefit fine-tuning on data from another setting. Here, we aim to show that a simple convolutional neural network can already be sufficient to extract features that generalize between domains. While more complicated and expensive models may allow for better performance, these additions are not strictly necessary for out-of-domain generalization. We pre-train a convolutional model on a sleep-staging dataset and show that it benefits learning on three downstream tasks: abnormality detection, Autism classification and classifying the effects of scopolamine. In all cases, we show faster convergence than a baseline trained from scratch, reaching similar performance. Additionally we show that for some of these tasks the pre-trained representations (before fine-tuning) capture the class annotation well enough that they are clearly visible in low-dimensional visualizations.

Keywords: EEG · self-supervised learning · generalized models.

1 Introduction

An electroencephalograph (EEG) is a recording of electrical potentials produced by the brain, recorded by placing various electrodes on the scalp [13]. Individuals doing different tasks will produce relatively similar brain activity which is reflected in EEG recordings. A central problem in the analysis of EEG recordings is that recordings made in different contexts can differ in superficial ways. Here, we refer to these details of the recording context as the *domain*.

Because of superficial differences, the features extracted from an EEG signal in one domain do not easily transfer to another. However, recent results show that certain deep learning methods, specifically the technique of *self-supervised learning* (SSL), allows for the transfer of meaningful features between domains [9].

In SSL, the structure of a large, unlabeled dataset is used to turn an unsupervised learning problem to a supervised learning problem, called a pretext or *pre-training task* [8]. A pre-training task is a task predicting surrogate supervisions defined on unlabeled data [18]. Surrogate supervision is created from the data itself by obscuring certain elements of the data, using the obscured elements as the prediction label.

The most notable benefit of using self-supervised learning techniques is that they do not require annotated data for the pre-training task. Annotated data is expensive and prone to human error and bias. Self-supervision potentially allows for learning more general and robust features than supervised learning models [14]. Neural network models learn representations following this pre-training task, for which this learned representation is used to solve a target problem called the *downstream task*. In effective SSL settings, the pre-training often causes the model to converge faster on the downstream task, and possibly also to a higher level of accuracy.

In [9] it was shown that a deep learning model can be used to learn a feature-extractor for EEG data whose features transfer between domains. Here, the model in question is a mixture of convolutional and self-attention layers. In earlier research [3], it was shown that a simpler, purely convolutional architecture could effectively perform self-supervised learning, but the model was only evaluated within-domain.

Our research question is whether the second, simpler model also allows for some measure of domain transfer. That is, can it be pre-trained on one domain, and benefit learning on a data in another domain?

To answer this question we pre-train a convolutional neural network on a sleep-staging dataset to extract a learned feature representation, then obtain feature vectors (embeddings) from other datasets for which a supervised model is trained in order to solve a downstream task. We compare these results against a fully-supervised convolutional neural network to establish whether some *general knowledge* about EEG has been learned by comparing early accuracy and speed of learning with training examples used. We also make use of dimensionality reduction techniques to visualize any identifiable clusters of classes induced by the SSL pre-training task.

2 Related Work

EEG is notoriously difficult to interpret, since it is hard to distinguish task-relevant information from more superficial features [15]. Despite this fact, the electrical signals making up EEG data is still governed by some physical and physiological laws with distinguishable patterns [7]. SSL can be used to learn a representation of this data by relying on the general structural features found, unhindered by the complexity of such a signal. Indeed, multiple authors claim high accuracies using SSL on sleep-staging EEG data (76.66% with *relative positioning* [3] and 88.16% with *contrastive learning* [7]), while reducing the amount of annotated data required. [9] adapted a language model originally designed for automatic speech detection and capable of processing enormous amounts of audio data to instead handle EEG data. Through a process of SSL, [9] encoded EEG segments as a sequence of learned feature vectors which they call *BErt-inspired Neural Data Representations* (BENDR), *BERT* referring to the adapted language model (Bidirectional Encoder Representations from Transformers). This representation was used to fine-tune target models and achieved comparatively high accuracy, outperforming prior sleep stage classification performance.

3 Materials and Methods

This project is based extensively on the research of [3,2]. Thus, most of the techniques and tools that are used are the same, with some exception where we involved functionality to allow new datasets to be processed and to allow for evaluation of baselines and SSL models, and plotting functions.

Specifically, we take from [3,2] the model architecture and the choice of sleep staging as a pre-training task.

3.1 Data

Sleep Physionet Dataset The pre-training dataset used is the Sleep Physionet EDF dataset containing 153 overnight sleep recordings from 83 healthy subjects of ages 25 to 101 [1]. Signals were recorded with EEG channels *Fpz-Cz* and *Pz-Oz* at 100 Hz. Data was annotated at 30-second intervals by experts following the Rechtschaffen & Kales (R&K) sleep-staging standard, with the exception of sleep staging 3 and 4 which were combined following the American Academy of Sleep Medicine (AASM) standard [4]. This resulted in a dataset with 5 class labels: Wake (W), Rapid Eye Movement (REM, R), Non-Rapid Eye Movement (NREM, N) 1-3. These class labels are not used during pre-training, but we do use them when finetuning the model on the sleep staging dataset itself.

TUAB Temple University Hospital (TUH) provides a set of public datasets which can be used for research and commercialization purposes. We used the TUH Abnormal (TUAB) [10] EEG Corpus as one of the datasets to perform our

evaluation on. TUAB contains 2,993 hours of adult EEG records from 2,383 subjects. Abnormality in EEG in this case was characterized by expert neurologists following a number of factors, including reactivity, α,β,μ,θ activities, and subjects' age and gender, all in relation to a person's state of consciousness (awake, drowsy or comatose) .

TUAB + white noise For the purposes of verifying our assumptions, we also include TUAB together with a class of white noise. This functions as a sanity check. If the proposed method works at all, the difference between white noise and any EEG data should be very apparent.

SPACE / BAMBI Dataset SPACE / BAMBI is a private dataset comprising individuals with Autism Spectrum Disorder (ASD), epilepsy, and healthy subjects as controls.[5] The dataset contains a large amount of artifacts which were cleaned prior to preprocessing and segmentation. EEG was recorded with 64 channels of which included *Fpz*, *Cz*, *Pz*, and *Oz*, selected and re-referenced as *Fpz-Cz* and *Pz-Oz*. Since the majority of recordings were of class ASD, a limiter was set in an attempt to mitigate the imbalance by taking minimum number of recordings of all three classes and regarding that amount as the limit of recordings to consider. A class imbalance was present nonetheless since cleaning artifacts of varied lengths will result in having recordings with different durations, thereby having different number of samples. To mitigate the issue, we also use a balanced accuracy loss while training.

Scopolamine Dataset Scopolamine is the most extensively used pharmacological model of cognitive impairment and was administered as a 15-minute intravenous infusion to 83 healthy male subjects aged 18–55 years in this dataset. 0.5 mg of Scopolamine (or placebo) was used to demonstrate that pharmalogical cognitive-enchancing compounds of well renown have significant effect in the cognition activity of healthy volunteers. *Fz*, *Cz*, *Pz*, *Oz* EEG channels were recorded and referenced as *Fz-Cz*, *Pz-Oz*. Recordings were performed over a period of 8.5 hours with 11 measurement time-points from baseline (pre-dose), each recording lasting 64 seconds. [16] states that peak scopolamine was found to be at measurement 03 (M03) and that therefore, this was considered to be the most distinctive. Selected time-points were M01 (baseline), M03 (peak), M05 (temporal median), M11 (wash-out). Only trails with scopolamine CHDR0507 (drug: R213129) and CHDR0511 (drug: R231857) were selected. We do not include the placebo class.

3.2 Preprocessing

First, the signal is converted from volts to micro-volts. Second, the EEG data is filtered using a band-pass filter in order to filter out signal frequencies which

[5] Obtained from the Center for Neurogenomics and Cognitive Research (CNCR), department of Integrative Neurophysiology (INF).

are lower than 0.5 Hz and higher than 30 Hz to filter out frequencies which are not critical to sleep-staging. This approach was also taken by *Chambon et al.* in the preprocessing stage [5]. Filtering is performed using *MNE* FIR filtering as implemented in the *Braindecode* framework.

Third, signals are split into non-overlapping segments of 5 seconds along with their respective annotations. In [3], annotations were made for 30 second segments. However, since the records from the downstream datasets are much shorter than the average 7 hour long Sleep Physionet recordings, we had to shorten the window length.

Finally, the windowed dataset is normalized using a channel-wise Z-score normalization. Table 1 lists the number of samples per class for each dataset that are used to train for the downstream tasks. Balanced accuracy is used as the metric for evaluating the training results since certain classes have more samples than the rest.

Table 1: Number of 5 second window samples per class for each dataset evaluation. Lowercase letters indicate **a**bnormal, **n**ormal, **h**ealthy, **e**pilepsy and **w**hite noise.

Sleep Physionet	TUAB	TUAB + WN	SPACE / BAMBI	Scopolamine
4709 (W)	12533 (a)	12468 (a)	2942 (h)	1896 (M01)
2928 (N1)	14301 (n)	14702 (n)	2631 (e)	1896 (M03)
14772 (N2)		37400 (w)	1907 (ASD)	1980 (M05)
3180 (N3)				1968 (M11)
5178 (REM)				

3.3 Pre-training

We follow the approach of [5,2] in all experimental detail with the following exceptions. To adapt to tasks where high-frequency features are more relevant we change the input frequency to 100Hz. For the sleep staging data, this is the native resolution. For the downstream tasks, we downsample to this resolution.[6] We change the window length to $T = 5$, and change the kernel sizes to 50 (half the sampling frequency) and remove all dropout layers. All other details of the architecture are taken from the original model. Datasets are split into training (60%), validation (20%) and testing (20%) sets. See Table 2 for a full description of the model.

[6] Using the method mne.io.Raw.resample in the MNE software [6], which implements a low-pass filter followed by nearest neighbor sampling.

Relative Positioning In [3,2], several pre-training tasks are proposed. Here, we focus on the task of *relative positioning*. In this task, two short subsequences of the EEG signal–called *windows*—are sampled, either close together, or far apart (see Figure 1). The task for the model is to predict whether a given pair of windows was close together in the original data or far apart. This provides a binary classification task requiring some insight into the data, for which no manual labeling is required.

We sample pairs of time windows $(x_t, x_{t'})$ with $x_t, x_{t'} \in \mathbb{R}^{Tf \times C}$, with T the window length in seconds, f the signal frequency and C the number of channels. With equal probability, the starting times t and t' of these windows are chosen to be either farther apart than some hyperparameter τ_{neg} or closer together than some hyperparameter $\tau_{\text{pos}} < \tau_{\text{neg}}$. The pair is labeled with $y = -1$ in the former case, and $y = 1$ in the latter. The task is to predict the label given the pair. We use $\tau_{\text{neg}} = 15$ min and $\tau_{\text{pos}} = 4$ min.

To sample these pairs, the sequence is first sliced into windows of length T, excluding the first 30 minutes. The first window x_t is chosen uniformly. A label is chosen with equal probability and the second window $x_{t'}$is then chosen uniformly from all windows that are sufficiently close to, or far away from, x_t. We sample 2,000 pairs from each record in the sleep staging data to make our pre-training dataset.

During training, both pairs are passed through the same feature extraction network (see Table 2). The extracted features are then passed to a classification head to predict y. After pre-training the classification head is discarded and only the feature extractor is retained. See [2] for further details.

This pre-training task is expected to work because time windows close in time are likely to share similar features, especially in the context of sleep-staging where sleep stages last between 1 to 40 minutes [1]. Therefore, windows positioned closer in time will likely share the same properties, while windows further apart are more diverse. By this approach, we hope to generate an appropriate representation of the data in which some generic EEG features are highlighted.[7]

3.4 Reporting and Visualization

If pre-training is successful, our feature extractor should encode physiologically relevant information in our learned representation in the resulting 25-dimensional feature vectors (even before finetuning). To verify this, we apply different dimensionality reduction techniques to reduce the number of dimensions to two or three dimensions, so that the data can be scatterplotted. If the method is successful, then instances with similar features, for instance, the same label, should cluster together.We perform Principal Component Analysis (PCA), t-distributed Stochastic Neighbour Embedding (t-SNE) [11] and Uniform Manifold Approximation and Projection (UMAP) [12].

[7] This approach is similar to the concept of Slow Feature Analysis (SFA) [17]

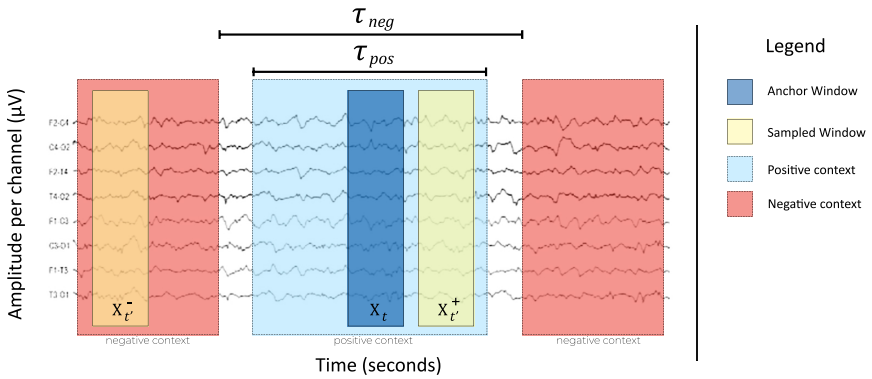


Fig. 1: Visual explanation adopted and edited from [2] showing how relative positioning works for a multivariate times series with EEG data. The figure describes the sampling process of selecting training examples in each pre-training task. Samples are selected by picking two pairs of windows randomly in a predefined range (τ_{pos} or τ_{neg}).

Table 2: The layers and layer specifications comprising our downstream classification model. This architecture largely follows [5,2]. Layers 3-10 are the layers in which feature extraction occurs. In pre-training the feature extractor is followed by a classification head which combines the features from both pairs.

	#	layer type	kernel	stride	padding	output	# params	activation
	1	input				(2, 500)		
	2	reshape				(2, 500, 1)		
feature extractor (embedder)	3	Conv2d	(2, 1)	(1, 1)	0	(1, 500, 2)	6	linear
	4	permute				(2, 500, 1)		
	5	Conv2d	(1, 50)	(1, 1)	(0, 13)	(16, 2, 477)	816	ReLU
	6	Batch Norm					32	
	7	Max Pooling	(1, 13)	(1, 13)	0	(16, 2, 36)		
	8	Conv2d	(1, 50)	(1, 1)	(0, 13)	(16, 2, 13)	12,816	ReLU
	9	Batch Norm					32	
	10	Max Pooling	(1, 13)	(1, 13)	0	(16, 2, 1)		
	12	Linear				25	825	

3.5 Downstream classification

To use the pre-trained model for classification on the downstream tasks, we pass samples from the pre-processed, segmented downstream data through the feature extractor, so that we obtain feature vectors. These feature extractors are then used to train a logistic regression model. Following [3,2], we keep the weights of the feature extractor frozen during downstream training.[8]

We compare the performance of the pre-trained model to a baseline model with the same architecture (as specified in Table 2) which has been randomly initialized and trained only on the downstream task. If the pre-trained model outperforms the baseline, we may conclude that there is some benefit to pre-training, even though the pre-training data came from a different domain.

We find that the benefit of pre-training appears most readily in the low-data domain: a model benefits most from pre-training when only limited downstream data is available. For each task, we train on a subset of N instances, and study the relation between N and the resulting accuracy. The idea is that even if the model does not outperform the baseline when all downstream data is available, we may still be able to see a benefit to pretraining in the low-data domain, which is sufficient to confirm our hypothesis that a convolutional network can, in principle, transfer physiologically relevant features between domains.

We first fine-tune the model using the pre-training data as a downstream task, as done in [2]. This serves to validate that the approach still functions after our minor changes. We then apply the model to the downstream tasks of Section 3.1.

4 Results

4.1 Pre-training

The pre-training network is used to obtain a model trained with sleep staging data for solving the relative positioning pre-training task. When pre-training on sleep staging data, we obtain a precision of 51%, a recall of 40% and an F1 score of 45% for the positive label $y = 1$. These results may not look promising, as they are close to random chance, but down-stream results and visualizations show that this is sufficient for effective self-supervision.

One reason for the reduction in balanced accuracy could be the changes made in the pre-trained network. The input was reduced to $\frac{1}{6}$ of the original size (5 second windows instead of 30 second windows), resulting in more samples and a greater chance of error. Layer parameters were set accordingly, and dropout was removed due to the low amount of output features.

[8] It may well be that fine-tuning the whole model, including the feature extractor, results in better performance. However, our aim here is to show that this model in its simplest and computationally least expensive form is already capable of generalizing between domains.

4.2 Plotting the Feature Space

In order to evaluate whether the learned representation obtained from performing self-supervision learning contains any trace of general EEG knowledge, a variety of datasets needed to be explored. We pass our three datasets (TUAB, SPACE/BAMBI and scopolamine) and the sanity check dataset (TUAB + white noise) through our pre-trained model to transform the raw data into corresponding feature vectors; n-dimensional vectors of numerical features representing the data.

We apply dimensionality reduction techniques PCA, t-SNE, and UMAP to reduce n-dimensional feature vectors to $n = 2$ or $n = 3$. For these plots, the model is not fine-tuned, and the feature vectors we extract come from a model that has never been exposed to the class labels of the downstream data.

We first apply this method to the sleep staging task that was also used for pre-training. Figure 2 shows six plots of three dimensionality reduction techniques to visualize how obtaining feature vectors through the pre-trained model compares to the raw segmented data. The plots show notable differences between their original counterparts, outlining some structure between the annotations. This reproduces results already shown by [3], but for our modified model. We also use additional dimensionality reduction methods and compare to a dimensionality reduction of the raw data, to show how much structure exactly is added by the model.

In Figure 3 we show UMAP plots for the downstream data. Figures (a) and (b) have the most clear structure. From these plots we can see that *abnormal*, *normal* and *white noise* signal classes are determined to be different enough for our pre-trained model to make a clear distinction. In figure (b), *white noise* (yellow) is exceptionally distinct from the *normal* and *abnormal* EEG features which functions as a sanity check as *white noise* and EEG data are significantly different from each other. In figure (c) (SPACE / BAMBI), *epilepsy* class (pink) is most prominent on the corners of the plot, but also scattered sporadically in the center, while *healthy* and *ASD* classes are scattered together with no evident disparity, suggesting that the the two classes may be similar.

We also plot the embeddings in a 3 dimensional feature space which allows for additional exploration of any possible clusters. Looking at figure 3 (d), the *M05* class seems to favour a particular side of the plot, meaning that some difference was indeed found from the rest of the classes. Figure 4 illustrates three different views of the same UMAP embeddings plotted in a 3D view where we can see clearer the *M05* class cluster residing in the plot.

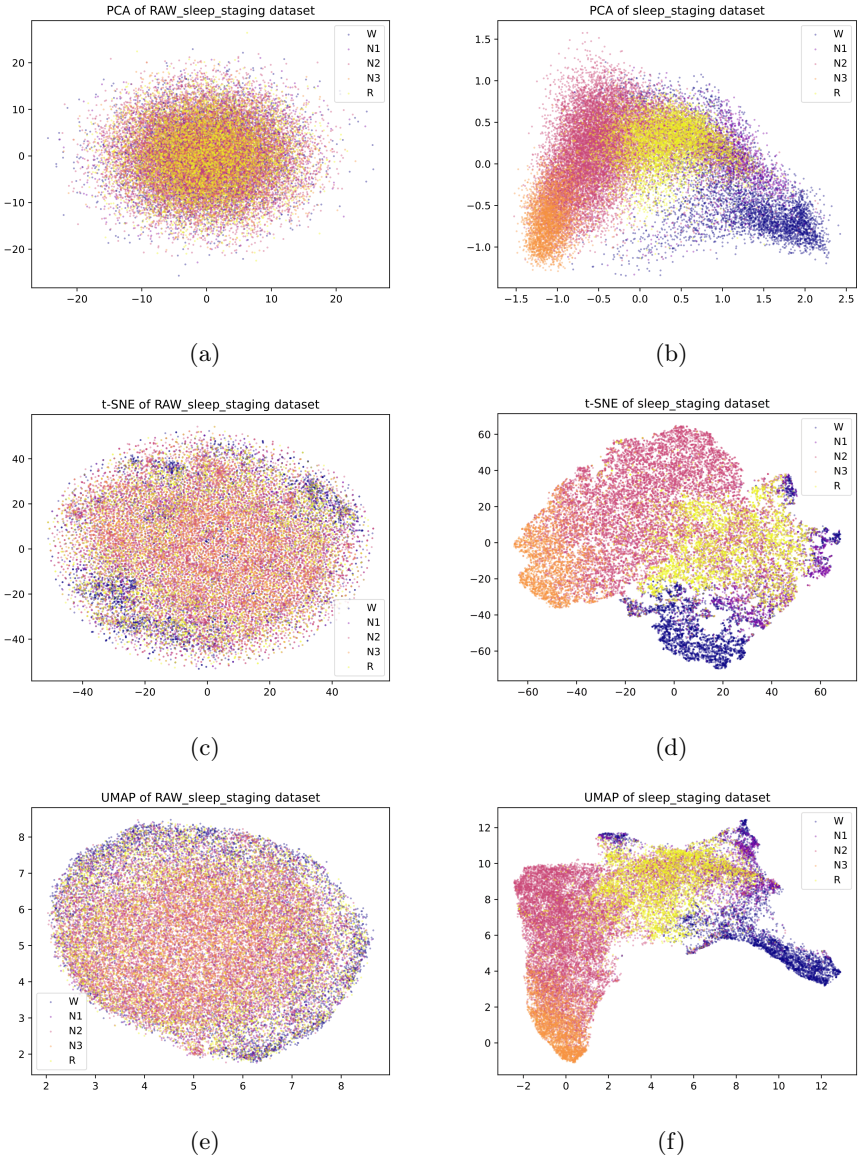
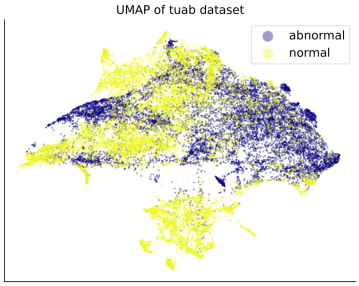
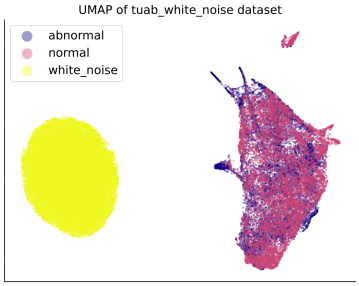


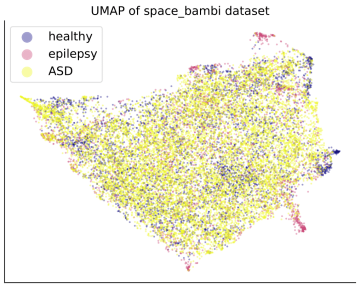
Fig. 2: PCA (a,b), t-SNE (c,d) and UMAP (e,f) scatter plots of 5-second window embeddings reduced from 25 dimensions to 2. Plots (a),(c) and (e) are scatter plots of 5 second window raw EEG data, while (b),(d) and (f) are scatter plots of the embeddings obtained through the pre-trained network. PCA, t-SNE and UMAP on the raw data do not capture any substantial class structure, but using feature vectors, the classes clearly cluster together.



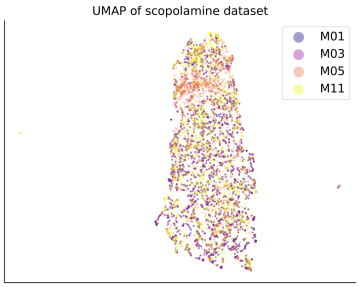
(a) TUAB feature vectors.



(b) TUAB + white noise feature vectors.



(c) SPACE / BAMBI feature vectors.



(d) Scopolamine feature vectors.

Fig. 3: (a) UMAP plots of 5 second window featuring TUAB, (b) TUAB + white noise, (c) SPACE / BAMBI, and (d) scopolamine datasets. Figures are UMAP representations of data obtained from feature vectors passed through the sleep-staging pre-trained model. The data visualized has been dimensionally reduced from 25 dimensions per 5 second time window to 2 dimensions. Note that in TUAB and SPACE/BAMBI, there is a clear separation of classes, even though the model used to produce these features was only trained on the sleep-staging data.

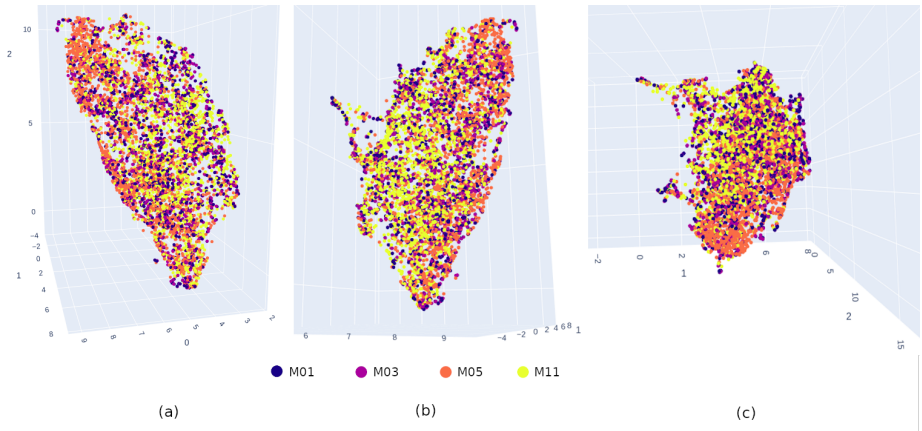
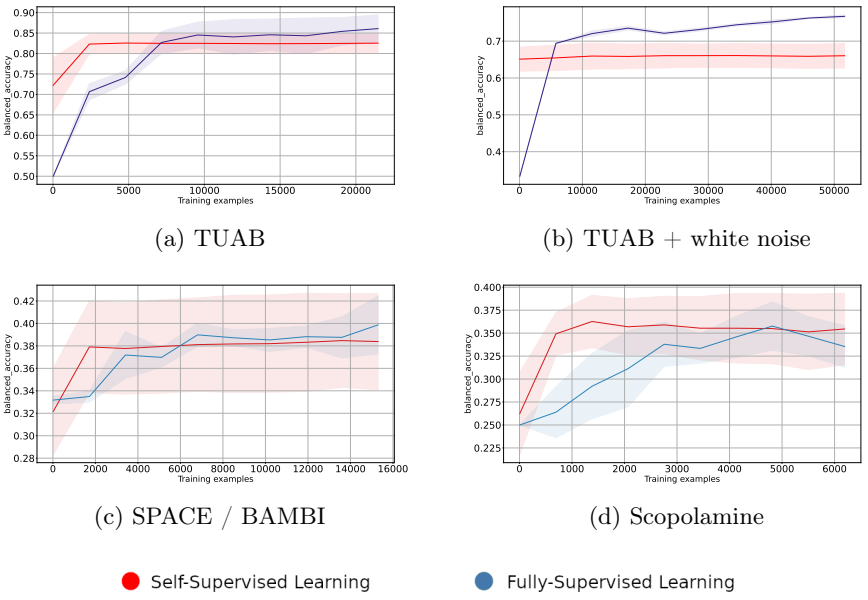


Fig. 4: UMAP of scopolamine dataset embeddings obtained from pre-trained model and plotted in 3D. Different perspectives make it easier to identify grouped class clusters.

4.3 SSL improves learning for EEG datasets in the low-data regime

We compare performance of the embeddings obtained from the sleep-staging pre-trained network to a Fully-Supervised Learning (FSL) convolutional neural network with the same architecture.

For each dataset, we fine-tune the pre-trained model on a subset of N instances from the training data on the downstream task. We compare against the same model without pre-training: that is, initialized randomly and trained directly, only on the downstream data. Figure 5 shows the resulting balanced accuracy plotted against N. In all cases, the SSL model shows a clear performance benefit in the low-data regime. While this is in some cases overtaken by baseline when more data is available, the results for the low-data regime suggest that some physiologically relevant information is transferred from the pre-training data to the downstream data. This is in line with the results from the visualization experiments.



(a) TUAB

(b) TUAB + white noise

(c) SPACE / BAMBI

(d) Scopolamine

Fig. 5: Plots of the balanced accuracy against training examples. Lines represent the mean obtained over a 5 fold cross-validation training run, while the opaque area represents standard deviation. In all cases, we see that self-supervised learning allows training to perform better in the low-data regime.

Table 3: Accuracies and classification of learning curves for Self-Supervised Learning (SSL) against Fully-Supervised Learning (FSL). SSL accuracies near convergences are higher than that of FSL, indicating faster learning with a smaller sample size.

	Training samples SSL near convergence	**Balanecd Accuracy**		**Training Samples** all	**Balanecd Accuracy**	
		SSL	FSL		SSL	FSL
Sleep Physionet W | N1 | N2 | N3+4 | N5	3000	**68%**	42%	25000	**69%**	65%
TUAB abnormal | normal	2500	**82.5%**	70%	22500	82.5%	**86%**
SPACE / BAMBI artifact | non-artifact | ignored	1000	**40%**	37%	6000	42%	**43%**
Scopolamine M01 | M03 | M05 | M11	1000	**36%**	27.5%	6200	**35%**	33%

5 Discussion

We used the *Sleep Physionet* EEG dataset to pre-train a convolutional neural network using self-supervised learning. We apply the knowledge learned from the representation using transfer learning to three different datasets (*TUAB*, *SPACE / BAMBI*, and *Scopolamine*) and additionally to *TUAB + white noise* for validation purposes. The visualizations of the feature space reveal structures of data classes which indicate that the class information is to some extent contained in the features extracted by the model even before fine-tuning. Localized structures of the data indicating relevant physiological information are extracted cross-domain, further validating the usefulness of applying our pre-trained model on different datasets.

Training on different amounts of downstream data shows that SSL performs better in the low-data regime for each dataset as indicated in Figure 5. However, in instances where more data is available, SSL is comparatively worse in some cases.

Ultimately, we find that self-supervised learning using a purely convolutional model is indeed useful as some information that is relevant between domains is to encoded by the learned feature extractor.

With these results, we show that a simple convolutional neural network architecture is sufficient for extracting relevant features that generalize over different EEG domains. Moreover, we can say that it is not only sleep staging knowledge that is encoded in the learned representation, but also knowledge about EEG data pertaining to the TUAB, SPACE/BAMBI and scopolamine datasets, suggesting that knowledge about the general EEG signals seem to be indeed present in our generated pre-trained model.

5.1 Limitations and future work

It is important to keep in mind that the architectural details of our SSL and FSL models are optimized for sleep-staging EEG data, data which is characterized by distinct time, frequency and proportions per sleep stage [5]. High disparity between attribute specifications of data classes makes it easier for any deep-learning model to identify distinct features. It may well be possible to achieve better downstream performance by changing the details of the architecture to better suite a diverse range of tasks. Additionally, including different tasks in the pre-training data may also improve performance (an approach that was followed in [9]).

We have shown that fundamentally, out-of-domain generalization can be achieved with a simple convolutional model. We based our architecture on the model of [3]. As noted, out-of-domain generalization was demonstrated earlier using a more complicated transformer model in [9]. To get more insight into exactly which aspects of the latter model contribute to its performance, a more thorough ablation study of that model would be required, on the datasets and tasks used there.

A convolutional architecture like the one used here has the benefit that it scales linearly in the length of the input sequence, as opposed to a transformer-based model, which (in the most commonly used implementation) scales quadratically. Precisely what the benefits and downsides of this tradeoff are, and what they mean for EEG analysis in practice requires more investigation.

6 Acknowledgements

This research was supported by TESS (Tertiary Education Scholarships Scheme). We thank Geert Jan Groeneveld from Centre for Human Drug Research for sharing the EEG data with the Scopolamine intervention. We also thank Hubert Banville for the support and guidance provided in the implementation phase of this research.

References

1. Altevogt, B.M., Colten, H.R., et al.: Sleep disorders and sleep deprivation: an unmet public health problem (2006)
2. Banville, H., Chehab, O., Hyvärinen, A., Engemann, D.A., Gramfort, A.: Uncovering the structure of clinical EEG signals with self-supervised learning (jul 2020). https://doi.org/10.1088/1741-2552/abca18, https://iopscience.iop.org/article/10.1088/1741-2552/abca18https://iopscience.iop.org/article/10.1088/1741-2552/abca18/meta
3. Banville, H., Moffat, G., Albuquerque, I., Engemann, D.A., Hyvarinen, A., Gramfort, A.: Self-Supervised Representation Learning from Electroencephalography Signals. In: IEEE International Workshop on Machine Learning for Signal Processing, MLSP. vol. 2019-October. IEEE Computer Society (oct 2019). https://doi.org/10.1109/MLSP.2019.8918693

4. Berry, R.B., Brooks, R., Gamaldo, C.E., Harding, S.M., Marcus, C., Vaughn, B.V., et al.: The aasm manual for the scoring of sleep and associated events. Rules, Terminology and Technical Specifications, Darien, Illinois, American Academy of Sleep Medicine **176**, 2012 (2012)
5. Chambon, S., Galtier, M.N., Arnal, P.J., Wainrib, G., Gramfort, A.: 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 (apr 2018). https://doi.org/10.1109/TNSRE.2018.2813138
6. Gramfort, A., Luessi, M., Larson, E., Engemann, D.A., Strohmeier, D., Brodbeck, C., Parkkonen, L., Hämäläinen, M.S.: Mne software for processing meg and eeg data. Neuroimage **86**, 446–460 (2014)
7. Jiang, X., Zhao, J., Du, B., Yuan, Z.: Self-supervised Contrastive Learning for EEG-based Sleep Staging. In: Proceedings of the International Joint Conference on Neural Networks. vol. 2021-July (2021). https://doi.org/10.1109/IJCNN52387.2021.9533305, https://github.com/XueJiang16/ssl-torch
8. Jing, L., Tian, Y.: Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (feb 2019), https://arxiv.org/abs/1902.06162v1
9. Kostas, D., Aroca-Ouellette, S., Rudzicz, F.: Bendr: using transformers and a contrastive self-supervised learning task to learn from massive amounts of eeg data. arXiv preprint arXiv:2101.12037 (2021)
10. Lopez, S.: Automated interpretation of abnormal adult electroencephalograms (2017), http://hdl.handle.net/20.500.12613/1767
11. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
12. McInnes, L., Healy, J., Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018)
13. Nunez, P.L., Srinivasan, R.: Electric Fields of the Brain: The neurophysics of EEG. Oxford University Press, USA, 2 edn. (2009). https://doi.org/10.1093/acprof:oso/9780195050387.001.0001
14. van den Oord, A., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., Kavukcuoglu, K.: Conditional Image Generation with PixelCNN Decoders. Advances in Neural Information Processing Systems pp. 4797–4805 (jun 2016), http://arxiv.org/abs/1606.05328
15. Schirrmeister, R.T., Springenberg, J.T., Fiederer, L.D.J., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., Burgard, W., Ball, T.: Deep learning with convolutional neural networks for EEG decoding and visualization. Human Brain Mapping **38**(11), 5391–5420 (nov 2017). https://doi.org/10.1002/HBM.23730
16. Simpraga, S., Alvarez-Jimenez, R., Mansvelder, H.D., Van Gerven, J.M., Groeneveld, G.J., Poil, S.S., Linkenkaer-Hansen, K.: EEG machine learning for accurate detection of cholinergic intervention and Alzheimer's disease. Scientific Reports **7**(1), 1–11 (2017). https://doi.org/10.1038/s41598-017-06165-4, http://dx.doi.org/10.1038/s41598-017-06165-4
17. Wiskott, L., Sejnowski, T.J.: Slow feature analysis: Unsupervised learning of invariances. Neural Computation **14**(4), 715–770 (apr 2002). https://doi.org/10.1162/089976602317318938, http://direct.mit.edu/neco/article-pdf/14/4/715/815111/089976602317318938.pdf

18. Yamaguchi, S., Kanai, S., Shioda, T., Takeda, S.: Multiple pretext-task for self-supervised learning via mixing multiple image transformations. arXiv preprint arXiv:1912.11603 (2019)