Evaluating Multitask Learning for EEG Denoising and Classification using Synthetic Data

Renato B. M. L. Rodrigues
School of Electrical and Computer Engineering (FEEC)
Universidade Estadual de Campinas (UNICAMP)
Campinas, SP, Brazil
renatobmlr@gmail.com

Denis G. Fantinato
School of Electrical and Computer Engineering (FEEC)
Universidade Estadual de Campinas (UNICAMP)
Campinas, SP, Brazil
denisf@unicamp.br

***Abstract*—When developing Electroencephalography (EEG)-based Brain-Computer Interface (BCI) systems, it is important to identify meaningful attributes from the EEG signals that could be used to train reliable classifiers and accurately decode brain patterns to control external devices. Multitask Learning (MTL) is a machine learning technique that can be employed to address this challenge, by simultaneously performing two or more tasks, such as denoising EEG signals and learning latent representations in order to train a classifier to predict different brain states. In light of this, in the present work we have created a Machine Learning (ML) model inspired on EEGNet capable of denoising and classifying EEG signals and a framework to generate synthetic data for evaluation. Results suggest that the MTL model was able to properly denoise and classify the EEG signals. The similar classification performance obtained using Single-Task Learning (STL) on a noisy version of the signal, when compared with the MTL model used to classify and denoise EEG signals suggest that the temporal and spatial convolutional filtering blocks may be effectively mitigating the impact of the noise.**

***Index Terms*—EEG, Brain-Computer Interface, Motor Imagery, Deep Learning.**

I. Introduction

When developing Electroencephalography (EEG)-based Brain-Computer Interface (BCI) systems, one of the biggest challenges is to accurately extract meaningful representations of the different brain states as well as to accurately discriminate task-related attributes from EEG artifacts [1]. Most part of the BCI systems use Single-Task Learning (STL) to classify the brain signals, and have a pre-processing stage in which non-brain components are removed using signal processing techniques such as Independent Component Analysis (ICA), among others [2].

Multitask Learning (MTL) is a powerful approach that allows multiple related tasks to be learned jointly, leveraging shared information and dependencies among them [3]. One of the key challenges when employing MTL is to determine when related tasks can be combined and what information can be shared among them, along with devising methods for sharing this knowledge effectively [4]. Conflicting tasks may give rise to a destructive inference phenomenon [5], wherein enhancing the performance of one task could result in degradation for the others.

There are considerably few studies exploring the usage of MTL for EEG-based BCI systems. These studies generally use MTL to: (*i*) reduce the subject variance, by considering each dataset as a different task and training a Machine Learning (ML) model to jointly decode the different brain states from all tasks [6], [7]; (*ii*) perform classification and regressions tasks by predicting an epileptic seizure event and its latency [8]; or (*iii*) simultaneously classify motor imagery patterns using a latent representation of the reconstructed EEG signal [9], [10]. In terms of the ML model's architecture, some studies are taking advantage of the sequential characteristic of the EEG time series by using Recurrent Neural Networks (RNNs) such as Long Short-Term Memory (LSTM) [6] and using low-dimensional representation of the EEG signal, obtained from the latent space of an AutoEncoder (AE), which was trained to reconstruct the same version of the EEG input signal [9], [10].

In this work, differently from [9], [10], we aim to use the latent space of an AE by training it to both classify and decode a denoised version of the EEG signals. This would allow to measure the effectiveness of removing artifacts while also classifying the different brain states. The ML model architecture is inspired on the EEGNet model [11], which is widely used to develop BCI systems given its robustness and compact model's architecture. Hence, the proposed new model architecture allows to simultaneously denoise and classify EEG signals. In order to have more control over the denoising and classification tasks, a simulated dataset was created. The model was trained using using a supervised learning approach and it was used a hard-parameter sharing MTL architecture [5].

This work is structured as follows. Section II presents the designed model architecture, the training and evaluation strategy, and the framework developed to simulate an EEG dataset that mimics brain behavior during a Motor Imagery (MI) paradigm. Sections III and IV present the results and their discussion, including experiments comparing denoising and classification performance using STL and MTL approaches. Finally, Section V, contains the conclusions and outlines future research directions.

II. Methodology

A. Model Architecture

A supervised MTL learning approach was used to train a model capable of denoising and classifying EEG signals, concurrently. For both the classification and denoising tasks, the ML model uses a shared encoder: in the classification branch it's followed by a classifier-decoder, and in the denoising branch by a denoising-decoder, as shown in Figure 1.

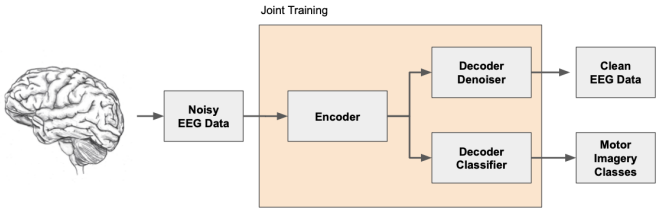

Fig. 1. Proposed methodology using Multitask learning.

The encoder's architecture was inspired on the EEGNet model [11], which performs spatial and temporal filtering through depthwise and pointwise convolutions. The encoder-model parameters are similar to the ones used in [11] considering a dataset with a sampling frequency of 128 Hz. The parameters used on the temporal convolutional block (Number of Filters - F1, Kernel size 1 - K1 and Pool size 1 - P1), with a deep multiplier D of 2, and, on the spatial convolutional block (Number of Filters - F2, Kernel Size 2 - K2, and, Pool Size 2 - P2) can be seen in Table I.

TABLE I
Encoder Parameters values.

F1	F2	D	K1	K2	P1	P2
8	16	2	(1, 64)	(1, 16)	(1, 8)	(1, 16)

It was used 0.5 as the dropout value, similar to the approach used by [11] on a within-subject evaluation.

On the denoising-decoder architecture, the same operations, but symmetrically, have been applied. It was explored an upsampling strategy using a linear interpolation, with the same scaling factor, as defined in the corresponding and symmetrical polling layers on the encoder. In order to improve the capacity of the model to reconstruct the original EEG signal provided as an input, it was included some skipped connections from the layers of the encoder to the decoder. This approach has allowed to forward some feature maps that have been learned on the encoder during the training phase [12], [13].

The classifier decoder consisted of a simple linear layer using as an input a flattening version of the feature maps obtained on the encoder.

B. Loss Function

When using STL separate models are trained independently to denoise and classify the EEG signals. Each model uses a distinct loss function: Mean Squared Error (MSE) for denoising and Cross-Entropy (Cross-Entropy) for classification.

By minimizing the MSE loss, given by:

$$\mathcal{L}_{\text{MSE}}(i) = \frac{1}{d} \left\| \mathbf{x}(i) - \hat{\mathbf{x}}(i) \right\|_2^2 \tag{1}$$

where $\mathbf{x}$ denotes the clean (denoised) version of the signal and $\hat{\mathbf{x}}$ is the output of the denoising decoder, the model aims to learn an optimal mapping function f^* that accurately reconstructs a denoised version of the signal from a noisy input for the ith-electrode – in a total of n electrodes.

For the classification task, the model approximates the mapping function f^*, which assigns brain states to model outputs, by minimizing the categorical Cross-Entropy loss:

$$\mathcal{L}_{\text{CE}} = -\sum_{c=1}^{C} y_c \log(\hat{y}_c) \tag{2}$$

where C represents the total number of classes, y_c is the ground-truth label encoded as a one-hot vector, and $\hat{y}_c$ is the predicted probability for class c obtained as an output of the classifier.

The main idea when using MTL is to leverage the shared representations and dependencies between the classification and denoising tasks, enabling them to benefit from each other during training. Since the MSE and Cross-Entropy loss functions have different scales and convergence behaviors, their respective weights were dynamically adjusted as part of the optimization process during MTL training. Similarly to the approach adopted by [14], [15], we adopt the MTL function defined as:

$$\begin{aligned} \min_{\Theta_1,\Theta_2} \quad & \frac{1}{2\sigma_1^2}\mathcal{L}_{\text{MSE}}(f_1(\mathbf{x}_{\text{noisy}};\Theta_1), x_{\text{denoised}}) + \log(1+\sigma_1^2) \\ & + \frac{1}{2\sigma_2^2}\mathcal{L}_{\text{CE}}(f_2((\mathbf{x}_{\text{noisy}};\Theta_2), y) + \log(1+\sigma_2^2) \end{aligned} \tag{3}$$

where $\mathbf{x}_{\text{noisy}}$ and $\mathbf{x}_{\text{denoised}}$ are the noisy and denoised versions of the EEG signal, y represents the brain states, and, σ_1 and σ_2 are trainable parameters that weight the contributions of the individual loss functions, given by $\mathcal{L}_{\text{MSE}}$ and $\mathcal{L}_{\text{CE}}$, which represent the mean squared error and categorical cross-entropy losses, respectively. The terms $\frac{1}{2\sigma_1^2}$ and $\frac{1}{2\sigma_2^2}$ work as uncertainty-based scaling factors, while the regularization terms $\log(1+\sigma_1^2)+\log(1+\sigma_2^2)$ penalize excessive uncertainty.

C. Training Experimental Protocol

To assess the model's generalization capacity, it was utilized Stratified K-Fold cross-validation with $k = 5$. This data partition method ensures that class distributions were preserved across splits. For each fold, the mean and standard deviation were computed on the training portion and used to normalize both the training and corresponding validation data. Each EEG electrode signal was normalized to obtain zero mean and unit variance.

As the optimization method, it was used Adam with a weigh decay of $1e^{-5}$ and a global learning rate of $1e^{-3}$, acting as a scaling factor. It was used cosine annealing scheduling strategy to speed-up the convergence during the early stages of training. Each stage of the model encoder, classifier and denoising-decoder used different learning rates values of $1e^{-3}$, $1e^{-4}$ and $1e^{-4}$, respectively. The ML model was trained for 300 epochs with a batch-size of 32 and an early stop after 20 epochs of no performance improvements on the validation set.

To increase the robustness of the training step against an imbalanced set of batches, it was adopted a weighting sampler strategy with some Data Augmentation (DA) transformations, being performed at the dataset batch level. The weighting strategy was utilized for oversampling the EEG epochs from the minority class, whereas the DA was used to provide slightly different version of the EEG data to the model, during the training step. On the DA transformations, two different approaches have been implemented, additive noise and time window shift [16], with an independent probability of being applied with 30% on each mini-batch. The additive noise transformation, was defined as being a Gaussian noise, being added to the mini-batch signal, with zero mean and 1% of the standard deviation of the corresponding EEG epoch per channel. The time window shift was defined as being a shift on the amount of data points corresponding to 0.2 seconds, for each epoch per EEG electrode.

All the experiments, to train and run the inferences, were executed using a NVIDIA Tesla K80.

D. Evaluation Strategy

The MTL approach will be evaluated by comparing the same evaluation performance metrics that were obtained independently using an STL approach.

1) Denoising Evaluation Metrics: In order to evaluate the performance of the AE to denoise EEG signals, it will be used Normalized Mean Squared Error (NMSE) and Correlation Coefficient (CC), as denoising evaluation metrics [17], [18].

The CC measures the similarity between y^* and $\hat{y}$ and is given by:

$$\text{CC} = \frac{C_{y^*,\hat{y}}}{\sqrt{C_{\hat{y},\hat{y}} \cdot C_{y^*,y^*}}} \tag{4}$$

where $C_{y^*,\hat{y}}$ is the covariance between y^* and $\hat{y}$, C_{y^*,y^*} and $C_{\hat{y},\hat{y}}$ are the covariance of y^* and $\hat{y}$ with themselves - variance of the signals, respectively.

The NMSE measures the overall deviations between y^* and $\hat{y}$, and is defined as:

$$\text{NMSE} = \frac{\|\hat{y} - y^*\|_2^2}{\|\hat{y}\|_2^2} \tag{5}$$

where $|\cdot|_2^2$ denotes the squared Euclidean norm and y^* and $\hat{y}$ represents the denoised signal and the AE's output, respectively.

2) Classification Evaluation Metrics: To evaluate the performance of the classification task, it will be used as classification evaluation metrics Precision, Recall, F1-score, Accuracy, and, the Cohen's Kappa score.

The accuracy is a measurement to evaluate the proportion of the classes that were correctly classified:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{6}$$

where TP and TN are the true positive and negative instances, respectively, whereas FP and FN are the false positive and false negative instances that have been misclassified.

The recall and precision metrics are defined by:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

The F1-Score is an harmonic mean between precision and recall, and is given by:

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

The Kappa coefficient measures the level of agreement between the output of the model and the manual annotator [19], [20]:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{9}$$

where p_0 is the empirical probability of agreement and is given by

$$p_o = \frac{\text{TP} + \text{TN}}{\text{Total Instances}}$$

and p_e is the hypothetical probability of chance agreement:

$$p_e = \frac{(\text{TP} + \text{FP})(\text{TP} + \text{FN}) + (\text{TN} + \text{FN})(\text{TN} + \text{FP})}{(\text{Total Instances})^2}.$$

E. Dataset Description

In order to have more control over the denoising and classification experiments, a simulated dataset was created. In the following, we provide details about the developed framework for its generation. Briefly, about 13 minutes of data, corresponding to 160 trials, with a sampling frequency of 128 Hz were created. Each trial had a duration of 5 seconds which corresponds to 640 points.

1) Brain Components generation: The simulated EEG dataset replicates brain activity during a MI experiment. It includes three event types: rest, "left", and "right", where the latter two correspond to motor imagery of the right and left hands, respectively.

During a MI event the μ/α rhythms [21] are prominent in the sensorimotor cortex region and in the occipital lobe [21], [22], as shown in Figure 2. When compared with a baseline activity, during a MI task there might be suppression of the μ band and desynchronization of the α band [22]. Simultaneously, there might be an increase of the β band in the sensorimotor cortex.

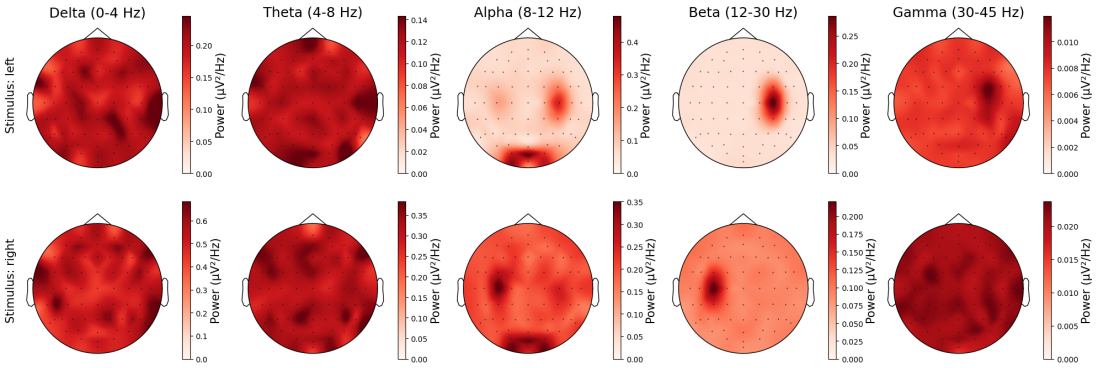

Fig. 2. Frequency bands from the simulated dataset for each stimulus: "Left" and "Right".

A dipole model was employed in order to mimic how electrical activity spreads across the different regions of the brain. The synthetic data generation model weighs the different signals by calculating the distance from each electrode to the centroid of the dipole, allowing for a more realistic representation of signal propagation in the brain. In order to approximate the spatial brain activity, it was assumed that activation of the regions of the brain follows a contralateral event principle [22], as illustrated in Figure 3.

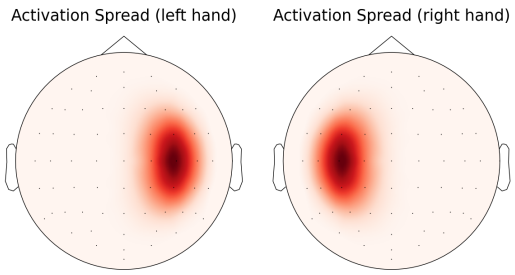

Fig. 3. Activation Spread for A) Left and, B) right hand movement. Dipole was considered to be on the electrode C4 and C3, respectively

To model brain oscillations, instead of the white noise, which has equal power across all frequencies, it was used pink noise which is characterized by a power spectral density that decays as $1/f$ [23].

2) Non-Brain Components generation: To simulate the artifact physiological signals, it was utilized NeuroKit, which is a Python Toolbox for Neurophysiological Signal Processing [24]. Two base synthetic signals were utilized, Electrocardiogram (ECG) signal, and, Electromyogram (EMG) signal, as shown in Figure 4.

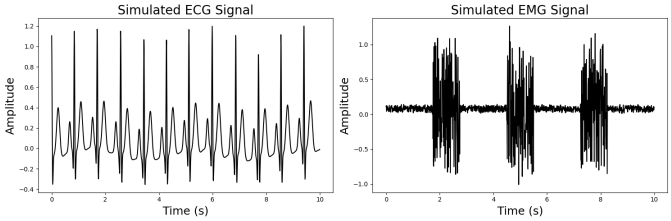

Fig. 4. Synthentic signals generated using NeuroKit [24]

3) Mixture of Components: The physiological signals corresponding to ECG and EMG were incorporated into the simulated EEG data, with spatial weights applied based on scalp topography. Specifically, ECG contributions were emphasized in the frontal electrodes, while EMG artifacts were weighted more heavily in the temporal regions. These spatial distributions are illustrated in Figure 5.

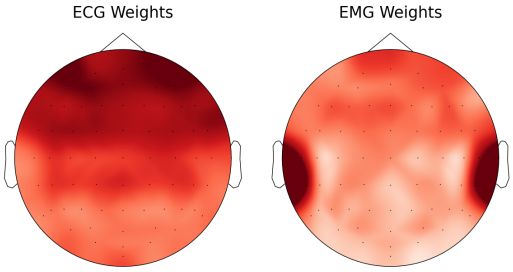

Fig. 5. Spatial weights used for the additive linear noise: (A) ECG, (B) EMG.

The resulting mixtures of brain and artifact signals were generated by linearly combining the sources using the defined spatial weights. Figure 6 shows an example of these mixtures for electrode C3.

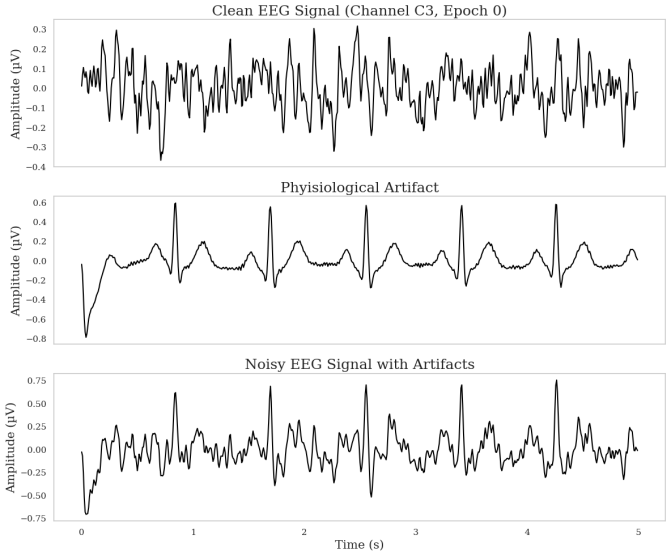

Fig. 6. Example of additive linear mixtures at electrode C3.

III. Results

The model's performance was evaluated by comparing the results on the denoising and classification tasks using STL and MTL approaches.

1) Denoising Performance Evaluation: As it can be seen in Figure 7, on the temporal and frequency representation of the noisy signal, it is possible to identify the EMG burst signal between 1.5 and 2.5 seconds, which was effectively removed in the denoised output of the AE.

The scatter plot of Figure 8 compares the denoised values with the ground-truth clean signal. We can observe a relatively high concentration around the ideal fit line with a low dispersion, which suggests that the model is satisfactorily reconstructing the clean signal.

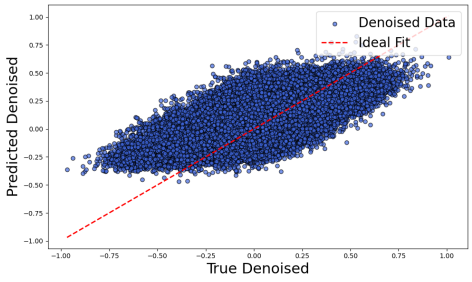

Fig. 8. Scatter plot with the Predicted and True denoised values using the MTL model.

In order to quantify the effectiveness of the denoising AE, it was calculated the CC and NMSE, as shown in Table II. The high value obtained for the CC and low values for the NMSE suggest that the model was able to effectively denoise the signals, demonstrating the model's effectiveness at denoising signals using STL and MTL, with STL achieving superior denoising performance.

TABLE II
Denoising performance metrics for STL and MTL classification

Metric	STL Denoising	MTL
CC	0.889 ± 0.001	0.745 ± 0.008
NMSE	0.209 ± 0.001	0.476 ± 0.016

2) Classification Performance Evaluation: As it can be seen in the confusion matrix and the ROC-curve from Figure 9, the model has shown to achieve a relatively high performance on the test set.

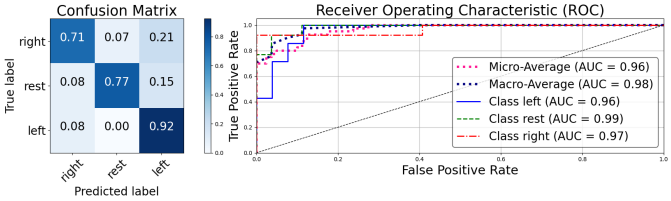

Fig. 9. Confusion Matrix and ROC-curve for the test set using the MTL model.

A comparison between the classification performance between STL and MTL can be seen in Table III.

TABLE III
Classification Performance metrics using STL and MTL

Metric	STL Classification	MTL
Accuracy	0.769 ± 0.011	0.779 ± 0.002
Precision	0.804 ± 0.011	0.797 ± 0.017
Recall	0.776 ± 0.010	0.784 ± 0.026
F1-Score	0.789 ± 0.001	0.791 ± 0.018
Kappa	0.656 ± 0.016	0.671 ± 0.032

IV. Discussion

The denoising performance evaluation using the CC and NMSE metrics suggests that both STL and MTL enabled the model to generalize and remove the physiological components introduced in the synthetic dataset.

With regard to classification performance, when comparing the average results across the 5 K-Fold validation, it was observed that MTL performed slightly better than STL. The Figure 10 shows a significant spectral overlap between the simulated brain activity and physiological noise. The classification results using STL on the noisy input signals suggest that the convolutional blocks of the encoder, responsible for temporal and spatial filtering, may be effectively mitigating the impact of the noise.

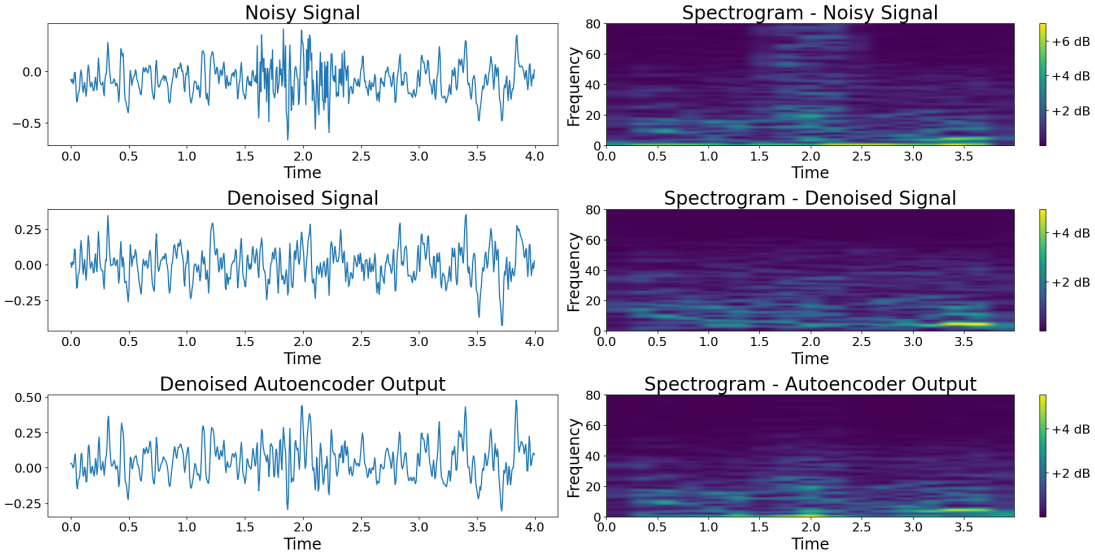

Fig. 7. Temporal and Frequency denoising comparison for the Test set on the Individual 1.

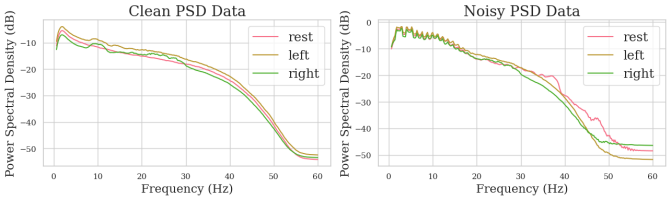

Fig. 10. Spectral comparison between the clean and noisy version of the signal for each stimulus.

V. Conclusion

In the context of BCI systems, we explored the use of MTL for denoising and classifying EEG signals using a simulated dataset. The results indicate that MTL has achieved a slightly better performance when compared with STL.

The similar classification performance obtained using STL on a noisy version of the signal, when compared with the MTL model that simultaneously classifies and denoises EEG signals, suggests that the temporal and spatial convolutional filtering blocks may be effectively mitigating the impact of noise. This also indicates that the two tasks are related, and that their combination does not result in destructive interference.

As next steps, we will evaluate the performance of the MTL method and the model architecture, (encoder + classifier) and (encoder + denoising decoder), on a set of publicly available MI datasets.

Acknowledgment

The author Denis Fantinato would like thank for grant 2023/00640-1, Sao Paulo Research Foundation (FAPESP).

References

[1] H. Altaheri, G. Muhammad, M. Alsulaiman, S. U. Amin, G. A. Altuwaijri, W. Abdul, M. A. Bencherif, and M. Faisal, "Deep learning techniques for classification of electroencephalogram (EEG) motor imagery (MI) signals: A review," *Neural Computing and Applications*, vol. 35, no. 20, pp. 14681–14722, 2023.

[2] A. Delorme, J. Palmer, J. Onton, R. Oostenveld, and S. Makeig, "Independent EEG sources are dipolar," *PloS one*, vol. 7, no. 2, p. e30135, 2012.

[3] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[4] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 12, pp. 5586–5609, 2021.

[5] M. Crawshaw, "Multi-task learning with deep neural networks: A survey," *arXiv preprint arXiv:2009.09796*, 2020.

[6] G. Panagopoulos, "Multi-task learning for commercial brain computer interfaces," in *2017 IEEE 17th International Conference on Bioinformatics and Bioengineering (BIBE)*, pp. 86–93, IEEE, 2017.

[7] A. Van Esbroeck, L. Smith, Z. Syed, S. Singh, and Z. Karam, "Multi-task seizure detection: addressing intra-patient variation in seizure morphologies," *Machine Learning*, vol. 102, pp. 309–321, 2016.

[8] X. Ma, S. Qiu, Y. Zhang, X. Lian, and H. He, "Predicting epileptic seizures from intracranial EEG using LSTM-based multi-task learning," in *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pp. 157–167, Springer, 2018.

[9] Y. Song, D. Wang, K. Yue, N. Zheng, and Z.-J. M. Shen, "EEG-based motor imagery classification with deep multi-task learning," in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2019.

[10] P. Autthasan, R. Chaisaen, T. Sudhawiyangkul, P. Rangpong, S. Kiatthaveephong, N. Dilokthanakul, G. Bhakdisongkhram, H. Phan, C. Guan, and T. Wilaiprasitporn, "MIN2Net: End-to-end multi-task learning for subject-independent motor imagery EEG classification," *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 6, pp. 2105–2118, 2021.

[11] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces," *Journal of neural engineering*, vol. 15, no. 5, p. 056013, 2018.

[12] L.-F. Dong, Y.-Z. Gan, X.-L. Mao, Y.-B. Yang, and C. Shen, "Learning deep representations using convolutional auto-encoders with symmetric skip connections," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 3006–3010, IEEE, 2018.

[13] C.-H. Chuang, K.-Y. Chang, C.-S. Huang, and T.-P. Jung, "IC-U-Net: a U-Net-based denoising autoencoder using mixtures of independent components for automatic EEG artifact removal," *NeuroImage*, vol. 263, p. 119586, 2022.

[14] L. Liebel and M. Körner, "Auxiliary tasks in multi-task learning," *arXiv preprint arXiv:1805.06334*, 2018.

[15] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings*

of the IEEE conference on computer vision and pattern recognition, pp. 7482–7491, 2018.

[16] E. Lashgari, D. Liang, and U. Maoz, "Data augmentation for deep-learning-based electroencephalography," *Journal of Neuroscience Methods*, vol. 346, p. 108885, 2020.

[17] S. Phadikar, N. Sinha, and R. Ghosh, "Automatic EEG eyeblink artefact identification and removal technique using independent component analysis in combination with support vector machines and denoising autoencoder," *IET Signal Processing*, vol. 14, no. 6, pp. 396–405, 2020.

[18] K. Zeng, D. Chen, G. Ouyang, L. Wang, X. Liu, and X. Li, "An EEMD-ICA approach to enhancing artifact rejection for noisy multivariate neural data," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 24, no. 6, pp. 630–638, 2015.

[19] M. L. McHugh, "Interrater reliability: the kappa statistic," *Biochemia medica*, vol. 22, no. 3, pp. 276–282, 2012.

[20] R. Artstein and M. Poesio, "Inter-coder agreement for computational linguistics," *Computational linguistics*, vol. 34, no. 4, pp. 555–596, 2008.

[21] C. Neuper, R. Scherer, S. Wriessnegger, and G. Pfurtscheller, "Motor imagery and action observation: modulation of sensorimotor brain rhythms during mental control of a brain–computer interface," *Clinical neurophysiology*, vol. 120, no. 2, pp. 239–247, 2009.

[22] C. S. Nam, Y. Jeon, Y.-J. Kim, I. Lee, and K. Park, "Movement imagery-related lateralization of event-related (de) synchronization (ERD/ERS): motor-imagery duration effects," *Clinical Neurophysiology*, vol. 122, no. 3, pp. 567–577, 2011.

[23] W. S. Pritchard, "The brain in fractal time: 1/f-like power spectrum scaling of the human electroencephalogram," *International Journal of Neuroscience*, vol. 66, no. 1-2, pp. 119–129, 1992.

[24] D. Makowski, T. Pham, Z. J. Lau, J. C. Brammer, F. Lespinasse, H. Pham, C. Schölzel, and S. H. A. Chen, "NeuroKit2: A Python toolbox for neurophysiological signal processing," *Behavior Research Methods*, vol. 53, pp. 1689–1696, feb 2021.