AMAL Project: Paper Reproduction

Brain Signal Generation and Data Augmentation with a Single Step Diffusion Probabilistic Model

SCIENCES SORBONNE UNIVERSITÉ

Nam Le Hai
Enzo Durand
Damien Dam

bit.ly/TrainFigs bit.ly/SourceGit

① Introduction

In this paper, the authors propose a method for generating EEG signals using diffusion probabilistic models. They also implement progressive distillation to improve the efficiency of the reverse sampling process. The generated signals serve as augmented data for EEG classification models, specifically EEGNet.

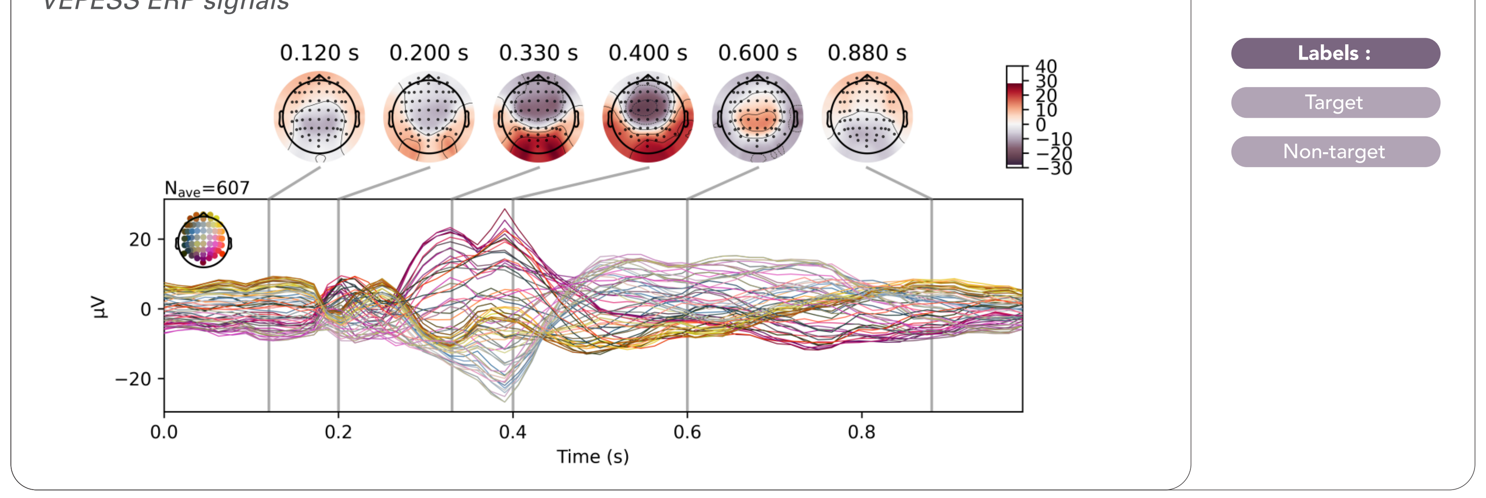
Contributions

/ Generate synthetic brain signals data using diffusion probabilistic models
/ Apply progressive distillation to speed up the reverse sampling process
/ Augment the brain signals dataset to improve accuracy in classification tasks

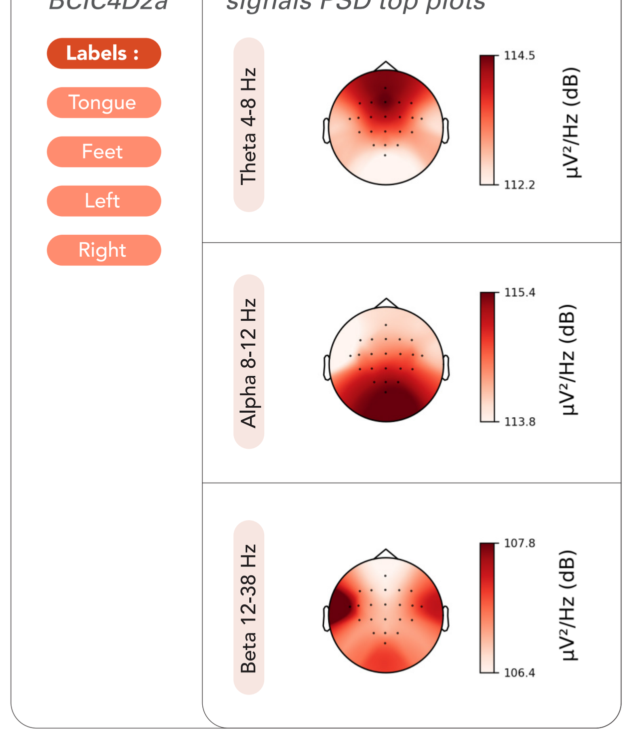
② Data

We work with 2 EEG datasets, **VEPESS** and **BCIC4D2a**.

VEPESS ERP signals



BCIC4D2a · *signals PSD top plots*



These datasets were collected from multiple subjects and feature recordings of subjects in response to triggering cues or instructed to imagine bodily movements. We divide the dataset into train, validation, and test partitions with a 70:15:15 ratio, stratifying by class labels and subjects.

③ Models

Continuous time model

We operate within the framework of a continuous-time diffusion model. The training loss is defined as a weighted denoising loss and can be understood as a weighted variational lower bound loss. During inference, we generate new data points using ancestral sampling.

Forward & reverse process

$$q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}(\mathbf{z}_t; \alpha_t\mathbf{x}, \sigma_t^2\mathbf{I}),$$
$$q(\mathbf{z}_t|\mathbf{z}_s) = \mathcal{N}(\mathbf{z}_t; (\alpha_t/\alpha_s)\mathbf{z}_s, \sigma_{t|s}^2\mathbf{I})$$
$$q(\mathbf{z}_s|\mathbf{z}_t,\mathbf{x}) = \mathcal{N}(\mathbf{z}_s; \tilde{\boldsymbol{\mu}}_{s|t}(\mathbf{z}_t,\mathbf{x}), \tilde{\sigma}_{s|t}^2\mathbf{I})$$
$$\mathrm{SNR}(t) = \alpha_t^2/\sigma_t^2 \quad 0 \le s < t \le 1$$

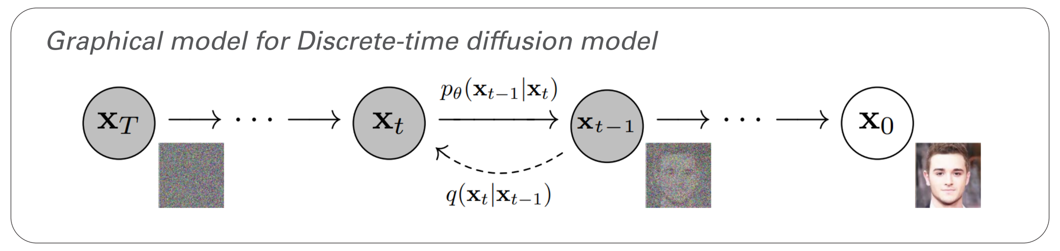
Training

$$\min_\theta L(\theta) = \mathbb{E}_{\epsilon,t}\left[\omega(\lambda_t)\left\|x - \tilde{x}_\theta(z_t,\lambda_t)\right\|_2^2\right]$$

Inference

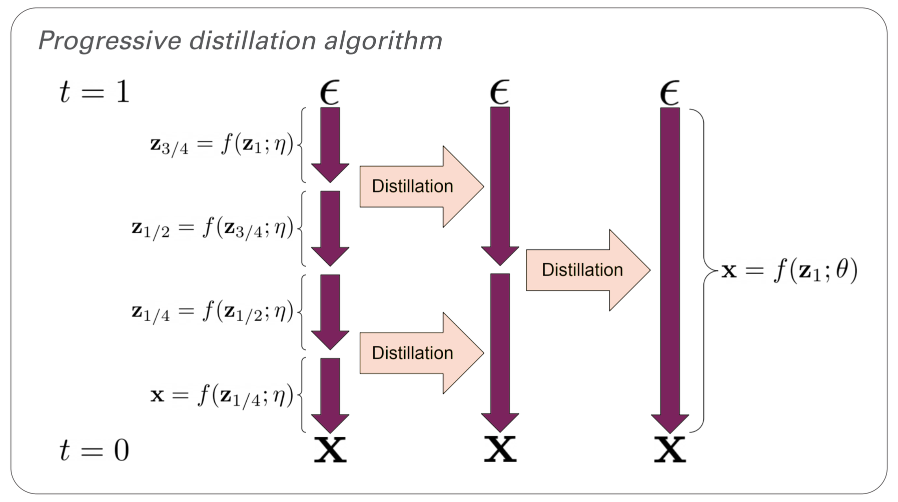
$$\mathbf{z}_s = \tilde{\boldsymbol{\mu}}_{s|t}(\mathbf{z}_t, \hat{\mathbf{x}}_\theta(\mathbf{z}_t)) + \sqrt{(\tilde{\sigma}_{s|t}^2)^{1-\gamma}(\sigma_{t|s}^2)^{\gamma}}\,\boldsymbol{\epsilon}\ ;\ \epsilon \sim \mathcal{N}(0,\mathbf{I})$$

Graphical model for Discrete-time diffusion model



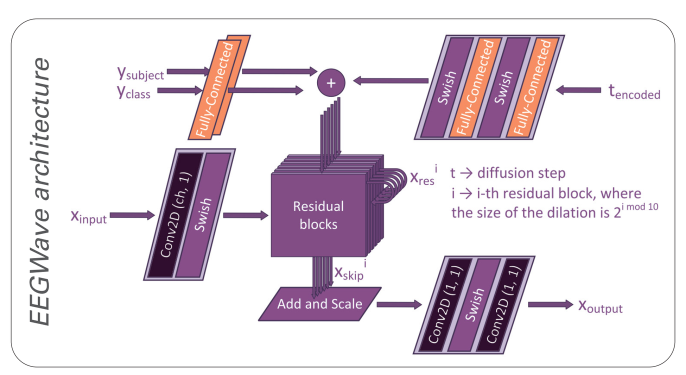
Progressive distillation

Progressive distillation algorithm



We use distillation to train a student model to generate latent variables through sampling with fewer steps compared to the teacher model. The goal is to have the student model denoise towards a target that aligns with two steps from the teacher's DDIM in just one step of the student's DDIM.

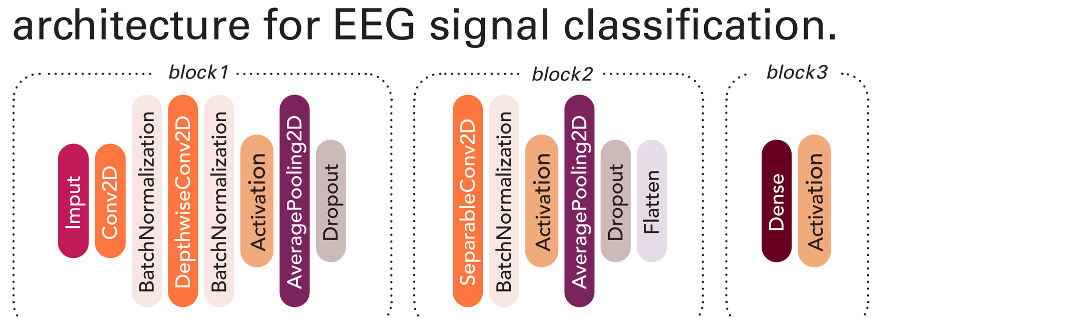
Function approximator



The denoising model $\hat{\mathbf{x}}_\theta(\mathbf{z}_t;t)$ predicting $\mathbf{x}$ from it's noisy version $\mathbf{z}_t$ is built using bi-directional dilated convolutions.

EEGNet

EEGNet is a compact and efficient deep learning architecture for EEG signal classification.



④ Results

Diffusion

We implement Inception Score (IS), Frechet Inception Distance (FID) and Sliced Wasserstein Distance (SWD). IS and FID are computed using representation produced by EEGNet.

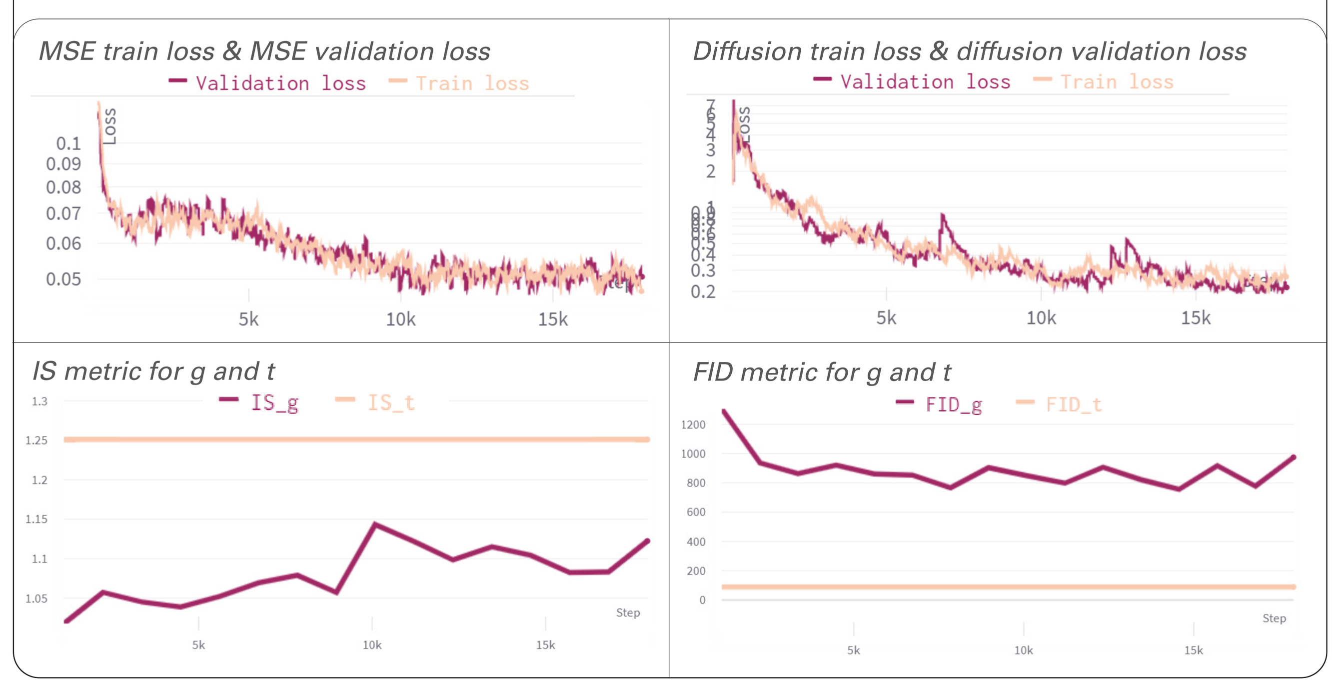
Class-conditional generation performance

		VEPESS			BCIC4D2a		
		IS ↗	FID ↘	SWD ↗	IS ↗	FID ↘	SWD ↘
Validation set		1.2375	14.64	43.72	1.2510	89.07	66.83
Generated	1024 steps	1.0086	450.02	1119.70	1.0279	783.97	740.88
	4 steps	1.0122	394.51	578.44	1.0138	973.86	214.25
	1 step	1.0253	346.63	908.26	1.0323	658.21	456.84

We compare the generated datasets to the test set. As a reference point, we compare the validation set to the test set.

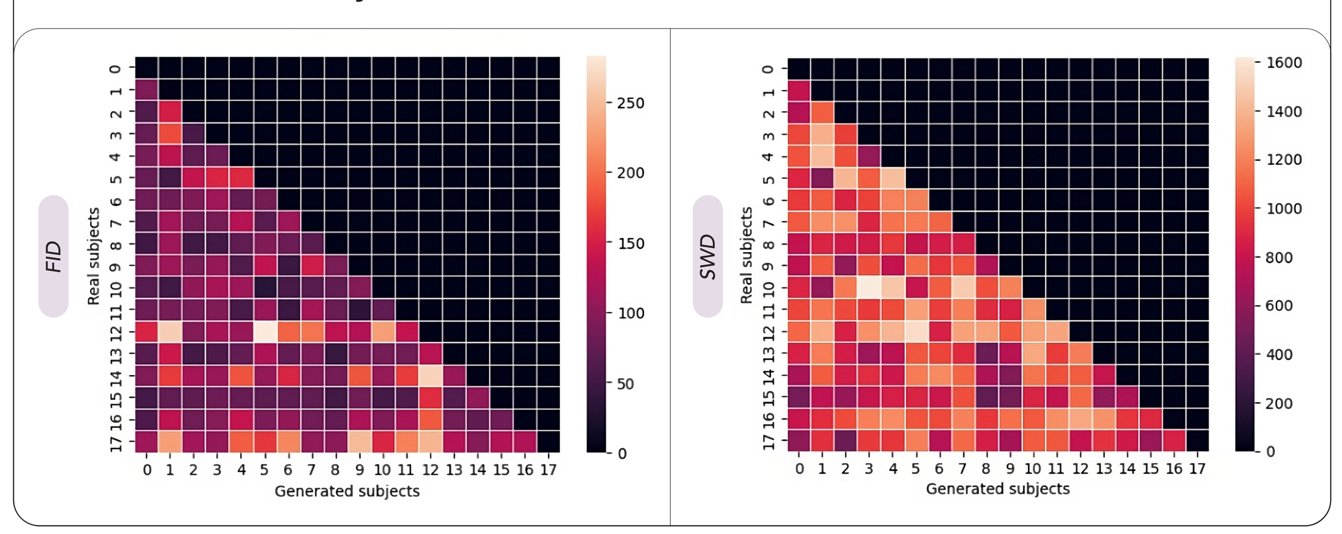
Learning progress

We display curves for various diffusion losses, as well as FID and IS metrics, throughout the training process.

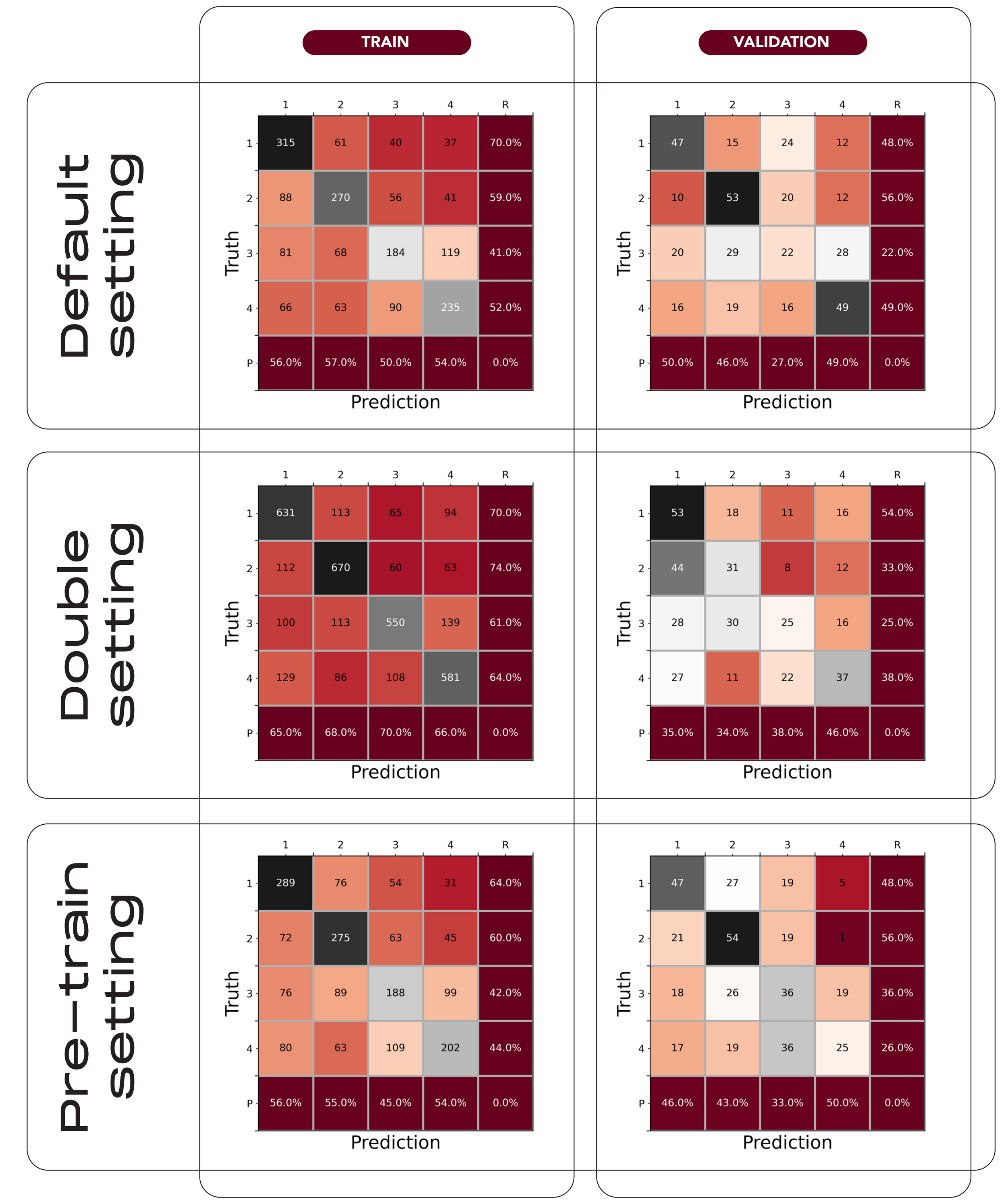


Subject specificity

We show heatmaps comparing generated signals conditioned on different subjects.



Augmentation



/ **Double setting** : Double the size by adding generated data.
/ **Pre-train setting** : Pre-train on the generated data, followed by training on the base training set.

⑤ Conclusion

The performance of EEGNet was not as good as reported when implemented using PyTorch, but improved when using TensorFlow. However, the diffusion model did not produce the expected results, leading to suboptimal performance in data augmentation tasks.

⑥ References

/ Salimans, Tim & Ho, Jonathan. *(2022). Progressive Distillation for Fast Sampling of Diffusion Models.*
/ Kingma, Diederik & Salimans, Tim & Poole, Ben & Ho, Jonathan. *(2021). Variational Diffusion Models.*
/ Jonathan Ho, Ajay Jain, and Pieter Abbeel. *(2020). Denoising diffusion probabilistic models.*
/ Kay Robbins, Kyung-min Su, W. David Hairston, *(2018). An 18-subject EEG data collection using a visual-oddball task, designed for benchmarking algorithms and headset performance comparisons.*
/ Lawhern, Vernon & Solon, Amelia & Waytowich, Nicholas & Gordon, Stephen & Hung, Chou & Lance, Brent. *(2016). EEGNet: A Compact Convolutional Network for EEG-based Brain-Computer Interfaces.*