ECoG-GAN: Data Augmentation for Electrocorticography

Yibo Jiao
University of British Columbia
Vancouver, Canada
jyibo@cs.ubc.ca

Kaseya Xia
University of British Columbia
Vancouver, Canada
zxia@ece.ubc.ca

ABSTRACT

Brain computer interface research has not created many standard frameworks recently mainly due to the lack of open public brain signal dataset. This paper introduces data augmentation on electrocorticography (ECoG) using General Adversarial Networks (GANs). We provide statistical evaluations on the quality of generated ECoG data by some state-of-the-art GANs. We also present our own novel attentive context normalization GAN (ACNGAN) with additional speech classification evaluation. Our ACNGAN performs better than any other GANs we tested and generates ECoG data that contains meaningful syllables.

KEYWORDS

Human Computer Interaction, Brain Computer Interface, CNN, Deep Learning, Electrocorticography, Generative Adversarial Networks

ACM Reference Format:
Yibo Jiao and Kaseya Xia. 2021. ECoG-GAN: Data Augmentation for Electrocorticography. In *Proceedings of CPSC 554X '21W1: Signal ML Course (CPSC 554X '21W1).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Brain-computer interface (BCI) systems are designed to connect brain to external electronics for various applications. The most common collected brain signals are electroencephalography (EEG) and electrocorticography (ECoG). EEG signals reflect abnormalities or other functional activities in the brain. By amplifying and decoding the electrical signal provided with EEG, researchers achieved brainwave controlled wheelchair for patients who can no longer walk [19]. EEG can be easily collected for patients in clinic or even at home with proper electrodes nowadays. But EEG do not present a scalable way to acquire data for numerous reasons. Examples are subjects needing to undergo long calibration sessions, classifiers trained on EEG datasets tending to generalize poorly to data recorded at different times, and cross channel interference leading to corrupted data.

ECoG is another way to measure brain activities; but ECoG is an invasive technique and provides higher temporal and spatial resolution compared to EEG. One major application of ECoG is to help patients who suffer from paralysis to communicate again based on their brain signal [18]. The real-time decoding of ECoG signal into high quality audio speech has been achieved by convolutional neural networks recently [1]. Nevertheless, the clinical ECoG data is even rarer than EEG due to its nature of invasiveness.

Due to the data-related problems such as lack of huge open-source public dataset and data corruption, BCI research has not been able to generalize its results to a wider scope. Rather than trying to improve the notoriously long calibration process to gather more real data, a more efficient way is to create synthetic data using Generative Adversarial Networks (GANs). The generation of artificial ECoG signals would have applications in many different areas dealing with decoding and understanding brain signals, but to our best knowledge no research regarding the generation of raw ECoG signals with GANs has been published at this time.

In this work, we apply GAN framework on ECoG with three major statistical evaluation methods to measure the quality of generated ECoG data: Inception Score (IS), Frechet Inception Distance (FID) and Root Mean Square (RMS). Since no one has done similar work before, we created our own baseline by implementing general GAN, deep convolutional GAN (DCGAN), Wasserstein GAN (WGAN), and Wasserstein GAN with gradient penalty (WGAN-GP). We first attempted time-series based approach following how synthetic EEGs are generated in [9]. But the training suffered from instability and model collapsing. So, we switched to 2D based approach by combining time-series with channels and achieved stable training outcomes for all GANs mentioned above. Inspired by the point cloud Attentive Context Normalization, we integrated DCGAN, WGAN and context normalization to build our novel Attentive Context Normalization GAN (ACNGAN). Our ACNGAN has showed superior performance compared with other state of the art GANs. Additionally, we also trained our own classifier to show that our ACNGAN can generate synthetic ECoG that contain meaningful syllables information.

2 RELATED WORK

2.1 GANs

GAN consists of two concurrent models: a generative model G that generates fake samples with latent noise input, and a discriminative model D that estimates the authenticity of a given sample with respect to the original dataset. Both generative model and discriminative model are deep neural networks that can be trained at the same time. The training goal for G is to maximize the probability of D making a mistake [7]. Over the training process, this two-player minimax game makes G generate better fake samples and makes D better at distinguishing fake and real samples [9].

DCGAN embeds deep convolutional layers in its generator and discriminator to perform unsupervised learning on the dataset.

CPSC 554X '21W1, Dec 17th, 2021, Vancouver, BC

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Training of this network enables learning of a hierarchy of representation so that similar distributed samples can be created [16]. DCGAN can generate realistic samples from noise once it is properly trained with hyperparameters that create a good balance between the generator and the discriminator. But one major challenge of DCGAN is that the training often collapses, where G cannot generate enough real samples for D to pass. Thus, the model could not converge and stays unstable. The recently proposed WGAN replaces the discriminator model with a critic that scores the realness of the input, which tends to improve the training stability [3]. But due to the weight clipping method in WGAN to impose the Lipschitz constraint on the critic, gradient could explode or vanish for some extreme weight clipping values. To solve this problem, a softer constraint called gradient penalty was proposed in [8] to further improve the training stability.

2.2 EEG GAN

Due to the promising results GANs can produce on images, researchers have attempted applying GANs to EEG in different applications. In [12], EEG signals were recorded from subjects while looking at images on a screen to reproduce the seen images. GANs outperformed variational auto-encoders with the help LSTM but the quality of the generated image still needs improving. Another study performed upsampling on EEG using GANs to interpolate the missing channel information [5]. Their results showed that GANs can generate more accurate data compared to bicubic interpolation.

More recently, [6] generated artificial EEG signals with DCGAN on 60 channels. Their analysis on real and fake EEG in both time and frequency domain indicated high resemblance. In [9], only one EEG channel was picked for generation using WGAN, but they provided a statistical baseline for measuring how close the artificial samples are comparing to the real ones. [11] further pointed out that the low level representation details could be lost when the GAN only aims to minimize the temporal error. They proposed a temporal-spatial-frequency loss function and showed better performance on EEG reconstruction for three motor-related EEG signal datasets.

2.3 ECoG to Speech

Researchers have been trying to decode brain signals from 20th century. But only recently more techniques and better results are showing up due to the prevalence of convolutional neural networks. In [10] , a linear regression model was used to decode ECoG into text. But the error rate was too high to be used in real applications. [15] used a Viterbi decoder with focus on spatial-temporal information, which significantly improved the accuracy. But this statistical model was slow and could not run real time. The use of 3D Convolutional Neural Networks (CNNs) by [1] showed high accuracy even with a small training dataset. Rather than directly decoding brain signals using CNNs, [2] used Recurrent Neural Networks (RNNs) to produce articulatory movements and then transform them into speech.

3 METHOD AND IMPLEMENTATION

3.1 Dataset

We used a public dataset [4], which contains 128 channel ECoG arrays implemented in a human patient during treatment for epilepsy. A total of 31 subjects were instructed to read aloud consonant-vowel syllables from a list while the ECoG signals were recorded at a sampling rate of 3k Hz. Each subject had between 256-289 syllables depending on the session length. The soundwave was recorded during the experiment but removed after for HIPAA compliance. Instead, detailed hand-marked annotations are provided with time step and the syllable name.

3.2 Signal Processing

We first resampled all channels to 500Hz for easier processing and filtering out the high frequency component. Then we used an IIR notch filter to attenuate the 60Hz line noise and the harmonics. To reference all the signals to a common ground, we estimated the mean over all channels and subtracted this quantity from all data points. Then we dissected out all the syllables from each subjects using a 128-sample points window along with their corresponding labels. The choice of this window length is for matching the channel size so that we have a 2D square input for our network. For the syllables that are shorter than 128-sample points, zeros were padded at the end.

We calculated logarithmic power on high gamma band (70-170Hz), which is known for containing the most amount of speech information [14]. We were mostly interested in high gamma band, but we also prepared the full signal in time domain to provide additional comparison. Both time domain and frequency domain signals were normalized to between -1 and 1. Eventually, we prepared three sets of data for training:

- Processed signal with all bands in time domain
- Gamma band signal in time domain
- Gamma band signal in frequency domain

The final output feature for each training set has size (W, C, S), where W (128) is the window length, C (128) is the channel size, and S (8856) is the syllables stacked from all subjects.

3.3 ECoG-GAN

We started implementing our network based on traditional GAN models. Our model evolved by adopting different components of axiomatic GAN models while running experiments on WGAN and DCGAN. We found that all traditional GAN frameworks are challenged by the nature of ECoG data, and we proposed a novel approach of combining context normalization and attention learning to solve drawbacks of original GAN models. After embedding our new block of layers in the modified pipeline, our method showed promising advances for adapting various types of ECoG data and training stability. In this section, we introduce our failed experiments and the solution we provided together with our contribution of integrating attentive context normalization with GANs.

3.3.1 Drawbacks of GAN. We initially implemented the very original GAN framework proposed by Ian Goodfellow [7]. During our first experiments, original GAN cannot generate any meaningful synthetic data, instead, the generator G only generates random noise. We ascribe the failure of original GAN to the instability and delicacy during training.

One of the drawbacks of GANs is the instability when training discriminator because the loss of G, D does not indicate the process of training. Due to the difficulty of extracting features from ECoG data, it is hard for us to monitor cases of mode collapse and synthesize meaningful signal data.

Our original GAN network tried to minimize the following loss function so that D classifies real distribution P_r as positive and generated distribution P_g as negative:

$$-\mathbb{E}_{x\sim P_r}\left[\log D(x)\right] - \mathbb{E}_{x\sim P_g}\left[\log(1-D(x))\right] \tag{1}$$

This made our D try to minimize Jensen-Shannon(JS) divergence between real ECoG signals P_r and synthetic ECoG P_g[7], when D reaches optimal cases $D^*(x) = \frac{P_r(x)}{P_r(x)+P_g(x)}$, we reformulate our objective function:

$$2JS(P_r||P_g) - 2log2 \tag{2}$$

Thus under (near) optimal D, if G generates a P_g that has negligible overlap with distribution P_r, this makes D training problematic because of gradient vanishing so that our G cannot learn and instead keeps generating random noises.

Another problem of this framework during experiments is mode collapsing. The loss function above has an unbalance penalty on diversity and accuracy of P_g. In other words, we put much more penalty on cases when G generates inaccurate distributions than generating similar or repetitive samples. Our D intends to collapse into distinguishing only few type of input distribution, which causes our G to only synthesize a single type of ECoG.

3.3.2 WGAN. Our solution for notorious problematic training processes of GANs is using Wasserstein distances. This method was initially proposed by Arjovsky et al[3], which provided us a more reasonable objective loss function and improved instability during training. In short, our WGAN implementation replaced original loss function by Earth-Mover(EM) distances:

$$W(P_r, P_g) = \inf_{\gamma\sim\Pi(P_r,P_g)} \mathbb{E}_{(x,y)\sin\gamma}\left[\|x-y\|\right] \tag{3}$$

where $\Pi(P_r, P_g)$ is the union of all possible joint distribution combining P_r, P_g, this is also called Wasserstein distances. EM distance advances JS divergences because even if there is no overlapping between P_r, P_g, this loss function still reflects the similarity between the two distributions. Thus our training process is indicated by Wasserstein distances and we solve collapse mode. In addition, we also applied the improved version of WGAN by adding gradient penalty, where we remove the weight clipping operation and still solve for Lipschitz continuity[3].

We implemented WGAN-GP before the milestone of this project, although our WGAN-GP generated multiple types of synthesis signals that are not random noise, there were apparent repetitive patterns in generated results. We thought the actual cause of these artifacts was the lack of well-designed structures of layers in WGAN-GP.

Our WGAN-GP implementation only consists of 3 layers of deconvolution to learn features from ECoG data. However, time-domain ECoG signals are noisy and lack of features; we require more elaborate networks to extract and learn features.

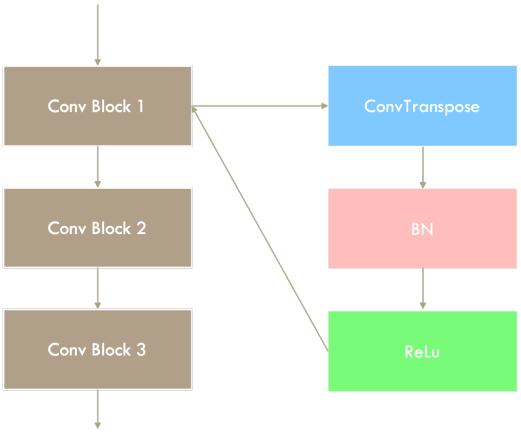

Figure 1: Our DCGAN generator model contains 3 convolution blocks for upsampling and learning features. In discriminator, we use similar structure but replace upsampling by downsampling.

3.3.3 DCGAN. WGAN-GP provides us improved objective function and a more reasonable way to optimize our network. The next step of our work is to embed CNNs into our model. In recent years, CNNs has been used widely in feature learning and we implemented DCGANs[16], which proposed a combination of deep convolution and GANs to further improve training stability and quality of results. Our DCGAN architecture is shown in Figure 1.

With DCGAN architecture and Wasserstein distances, our network had already been able to learn and generate meaningful signals that has reasonable statistical evaluation scores. To further improve the quality of synthetic signals, we contribute a solution that handles the noisy property of ECoG signals.

3.3.4 Context Normalization. To advance the quality of synthesization, we start by ablation studies of our DCGAN network. DCGANs highly relies on batch normalization, we removed BN layers in each convolutional block and evaluate performances of G. Experiments showed that withour BN, DCGAN generated nearly random noises. However, BN is very sensitive to outliers, with ECoG time-domain data that includes noises, we improve batch normalization by context normalization.

Context normalization[13], introduced a simple yet efficient operation that normalizes the features according to their distribution after every perceptron. This technique was originally used in point cloud learning and finding correspondences in epipolar geometry. We adopt the idea of normalization based on distribution and design a context normalization block for solving noisy cases of ECoG input.

3.3.5 Attentive Context Normalization. The intuition behind our novel method of attentive context normalization is that we want to find an automatic method to distinguish noise and real data. Since our method is already data-driven, we tried to learn the attention weights that can be used by context normalization.

We proposed a simple yet efficient operation that uses convolution to learn attentions, we call this attentive context normalization

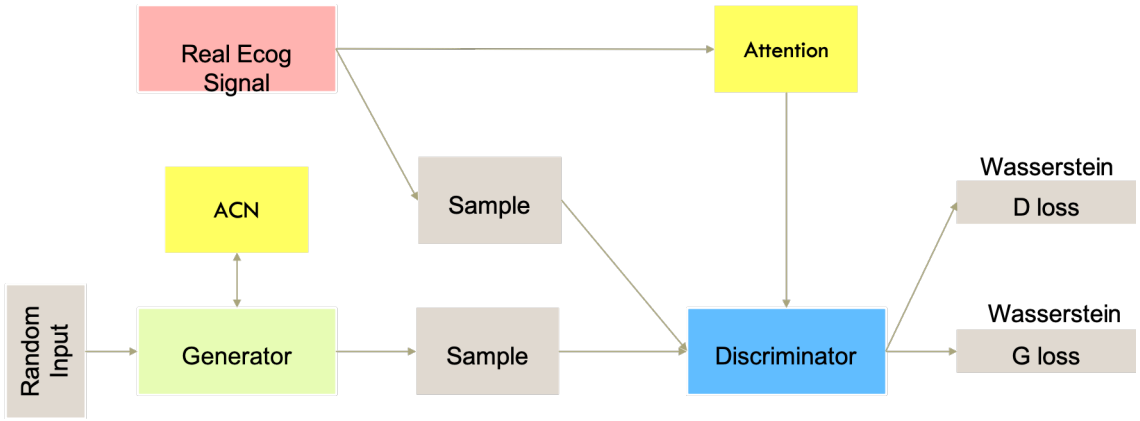

Figure 2: Pipeline overview of our network ACN-ECoGGAN, a method that combines attention learning, context normalization and DCGAN to synthesis ECoG data.

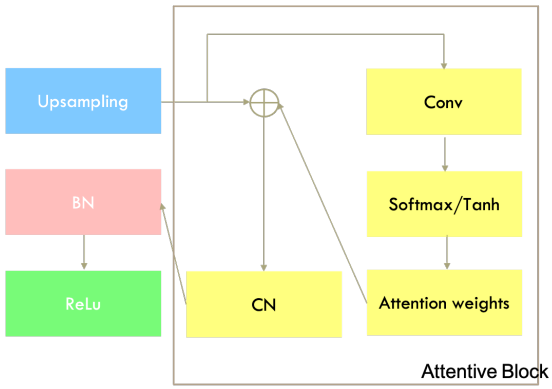

Figure 3: Our attentive context normalization block for attention learning.

blocks[20]. Our ACN block is shonw as Figure 3. We firstly take the output of convolution and pass them in our attentive block, it will firstly further convolve data and use an activation function such as softmax or tanh to activate, the output will be our learned weights. Along with the training progress, an attention weight will be learned so that the outliers(noises) have smaller weights and inliers(real data) have higher weights, with this attention, the context normalization works as simply a weighted normalization by calculating weighted mean and weighted standard deviation. Then we forward our attentive block output to batch normalization in order to improve synthesization quality.

3.4 ECoG to Speech Label Classification

For synthetic ECoG data, one way to evaluate the quality of generation is to predict their text labels with some classification network. Since our ECoG dataset has text labels for each time-domain signals. We train a VGG network with ground truth text labels for this classification task. Then we use the same trained VGG network to test our synthetic signals. Our method is to calculate a probability vector for each input synthetic signal that represents the likelihood for each text label. If the likelihood for the signal belonging to some text label are significantly large than all other labels, we are confident to say our generated ECoG is not only statistically similar to P_r but also in good quality to reconstruct text label information. The formula of the metric we use is:

$$v(x) = \text{softmax}(\text{sigmoid}(f_n \odot (f_{n-1} \odot (... \odot f_1(x))))) \quad (4)$$

$$p(x) = \|v(x)\|_\infty \quad (5)$$

where $f_1, ..., f_n$ are VGG convolutions, and we take the maximum probability. If $p(x) > 0.5$, our generated ECoG data is likely to be in one specific text label, and the generated ECoG is in good quality.

3.5 Pipeline

As described above, our network evolves from GAN, WGAN and DCGAN with our ACN block. An overview of our network is shown as in Figure 2.

Similar to traditional generator-discriminator GANs, our network consists of two concurrent model: G with ACN block and D with pre-trained attention.

As we mentioned in section 3.3.5, attention learning was applied in the generator, in order to make the discriminator D also take advantage of attention to distinguish noise and real data, we pre-train our discriminator with real ECoG signal to learn ground truth attention weights. We take all of our training ECoG signal as input and use exactly the same discriminator architecture to train truth attentions. For training data, we added different degree of extremities of noise on real data to generate noisy training data. We train the discriminator to classify input signals so that the more noisy the signal is, the lower classification score is. After pre-training, learned optimal weights are used in ACN blocks in discriminator D.

In generator G, we embed an ACN block in each layer of convolution for DCGAN based architecture, and we optimize Wasserstein distances as our objective loss function.

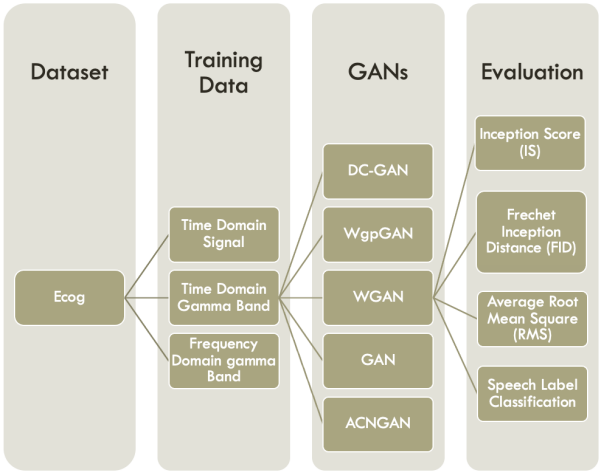

Figure 4: Training and evaluation pipeline overview: We ran experiments with 3 set of training data, 5 different GANs and evaluate with 4 evaluation methods

4 RESULTS

4.1 Statistical Evaluation Metrics

4.1.1 Inception Score. IS was first introduced in [17] to measure if an image belongs to certain class by an inception classifier. IS measures the quality of GANs' output by simultaneously looking at if the samples have variety and how realistic the sample looks like. In general, noise would have a low inception score, and high-quality samples would have a high inception score.

4.1.2 Frechet Inception Distance. FID is a more statistical way to measure how close the real samples and the generated samples are for GANs. FID also relies on the inception model to calculate the mean and covariance for real samples and generated sample, respectively. The distance between those two distributions is then calculated using Frechet distance. FID between real and itself would yield zero, between real and noise would yield a high score.

4.1.3 Root Mean Square. RMS is adapted from the course assignment 2, which measures the difference between the real samples and the generated samples. For the evaluation purposes, RMS was implemented as below: for each generated sample, each channel was selected and computed an RMS score with the corresponding channel of all original samples. Then all 128 channel RMS scores were averaged to produce the final RMS score. This implementation reflects the mean difference between real and generated samples. Real samples would have an RMS of 1, and noise would have an RMS close to 0. In the section below, we also present visual comparison between real and artificial samples with high RMS score.

4.2 FID, IS, RMS Scores

Figure 5, 6, 7 present FID, IS, and RMS scores for all architecture in full signal time domain, Gamma band time domain, and Gamma band frequency domain. Figure 8 presents FID score among 5 different GANs and 3 training dataset.

Model	FID	Inception	RMS
Noise	416	1.08	0.146
Real	0	1.36	1
DC-GAN	54	1.19	0.51
WGAN	174	1.4	0.534
WgpGAN	219.53	**1.59**	0.48
GAN	124	1.406	0.527
ACNGAN	**44.9**	1.22	**0.54**

Figure 5: FID, IS, RMS scores for time domain signal

Model	FID	Inception	RMS
Noise	416	1.08	0.146
Real	0	1.36	1
DCGAN	64	1.14	0.523
WGAN	89	1.21	**0.569**
WgpGAN	54	**1.36**	0.52
GAN	64	1.31	0.53
ACNGAN	**50**	1.14	0.51

Figure 6: FID, IS, RMS scores for Gamma band time domain signal

Model	FID	Inception	RMS
Noise	416	1.08	0.146
Real	0	1.36	1
DC-GAN	82	1.32	0.31
WGAN	117	1.22	0.373
WgpGAN	115	1.23	**0.67**
GAN	114.9	1.21	0.39
ACNGAN	**59**	**1.32**	0.307

Figure 7: FID, IS, RMS scores for Gamma band frequency domain signal

4.3 Signal Visual Comparison

Figure 9 and 10 present high RMS score real and fake sample comparison trained with full signal time domain and Gamma time domain datasets, respectively. Two different samples from different channels are showed in each figure.

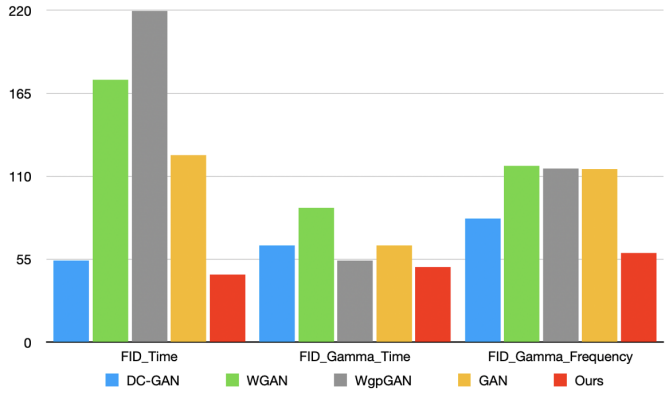

Figure 8: FID scores for full signal time domain, Gamma band time domain and Gamma band frequency domain

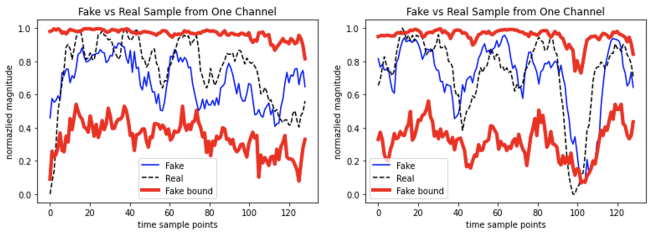

Figure 9: Time domain full signal training results comparison of one channel between real and fake samples. (Sample 1 Left: RMS = 0.82, Sample 2 Right: RMS = 0.83)

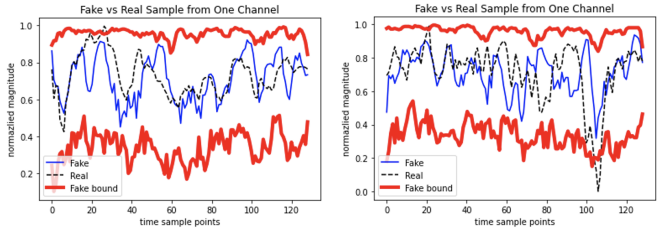

Figure 10: Gamma band time domain signal training results comparison of one channel between real and fake samples (Sample 1 Left: RMS = 0.87, Sample 2 Right: RMS = 0.83)

4.4 ECoG to Speech Label Classification

After we show artificial ECoG data similarity statistically and visually, we further present the feature representation quality in Gamma band for generated ECoG data. Note that since section 4.2 shows our network ACNGAN performs the best over other networks, we only used samples generated by ACNGAN to create the following evaluation results. We use each individual channel with varying sample size to train VGG and aim to achieve high accuracy. Figure 11 shows the accuracy of VGG trained with true ECoG data across all 128 channels. The higher accuracy value one channel has, the higher the probability that this channel contains more speech feature representations. In our results, channel #27-33 and #80-87 show relatively high accuracy value. Thus, we only select those channels for evaluating generated ECoG data quality. Figure 12 shows the training and validation accuracy for channel #81, which has the best accuracy results among all channels. Then we use ACNGAN generated signals with same selected channels as discussed above (#27-33, #80-87) to test synthetization quality, the result is shown in Figure 13.

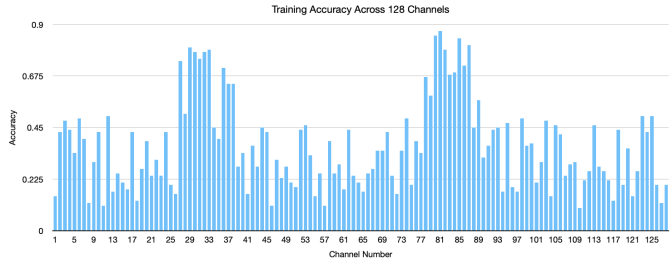

Figure 11: Training accuracy across all 128 channels using VGG-16

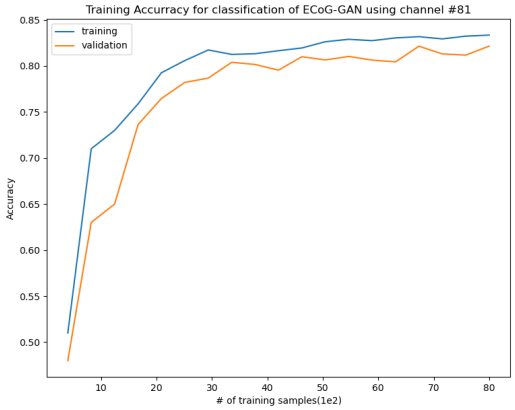

Figure 12: Training and validation accuracy for channel #81 using different amount of training data

5 DISCUSSION AND ATTEMPTED WORK

Inception score is highly sensitive to noise and is not able to detect mode collapse, as it solely relies on the final probabilities of the of the classifier [9]. RMS provides a very strict measurement since it even considers the phase shift of the whole signal. So, it could also vary due to the size of generated ECoG samples. But FID score is a more reliable method that provides a true evaluation of the quality of GANs. Even though Our ACNGAN was not the best for IS and RMS in all cases, it had the best performance in FID score for time domain signal, Gamma band time domain signal, and Gamma band frequency domain signal. The visual comparison in Figure 9 and 10 also gives us a good verification that our GANs have learned the major trending of the ECoG signal with enough details.

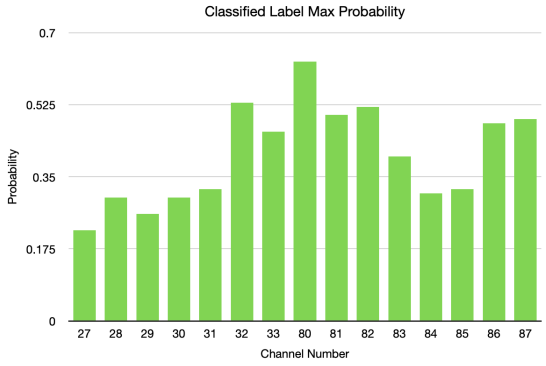

Figure 13: Classification max label probability among picked channels from synthetic ECoG. x-axis is selected channels, y-axis is the p-score metioned in equation 5

The speech audio was recorded at experiment time, but we didn't have access to it, which is a huge regret for this work. If we had the audio data, we would have followed the 3D DenseNet method as in [1] to create Mel-spectrogram for decoding the ECoG data into speech to evaluate our synthetic ECoG. Instead, our VGG classifier was trained to classify ECoG signal into labeled syllables with high accuracy. We started the classifier training using all 128 channels but got very low accuracy. We thought it is mostly because the small data size for too many feature extractions. Thus, we iterated our training loop for each individual channel to look for which channels capture most of the speech information. The results in Figure 13 shows that some channels (32, 80, 82, 86, 87) has a much higher p-score than other channels, which indicates our ACNGAN performed really well for generating synthetic speech features at those channels. We think a p-score higher than 0.5 is significant to prove that our ACNGAN generated a meaningful syllable since the second highest probability would already be lower than 0.5 in the same output vector (all syllable probability has to add up to 1). We cross-referenced those selected electrode positions and found they are mostly around inferior frontal gyrus pars opercularis, which correlates with the speech function area [1]. However, it is unknown what causes certain channels to attain higher p-score than others.

As suggested in [11] and [15], integrating spatial and temporal information will further improve the performance of the network. To add temporal information, we attempted to create multiple overlapping windows from the original signal and use 3D convolutional layers to build a 3D GAN. But the amount of work needed to finish this experiment exceeded our project scope. Future work can be done on this perspective and we would like to see how results compare to the baselines we provided in this work.

6 CONCLUSION

In this work showed that ECoG data can be augmented by different GAN structures. We created FID, IS and RMS baseline evaluation for GAN, WGAN, WGAN-GP, and DCGAN. After investigating the noise and outlier impact on GANs, we further developed our own novel ACNGAN, which showed superior performance than other GANs we tested. Eventually, our trained VGG classifier verified that our synthetic ECoG contains meaningful speech syllable information using selected channels.

7 CODE

Our code and implementation can be found here:
https://github.com/yibojiao/ECoGGAN.git

ACKNOWLEDGMENTS

We would also like to thank Dr. Robert Xiao, TA Abiramy Kuganesan, and other students in CPSC 554X for their feedback and suggestions throughout the project.

REFERENCES

[1] Miguel Angrick, Christian Herff, Emily Mugler, Matthew C Tate, Marc W Slutzky, Dean J Krusienski, and Tanja Schultz. 2019. Speech synthesis from ECoG using densely connected 3D convolutional neural networks. 16, 3 (apr 2019), 036019. https://doi.org/10.1088/1741-2552/ab0c59

[2] Gopala Krishna Anumanchipalli, Josh Chartier, and Edward F. Chang. 2019. Speech synthesis from neural decoding of spoken sentences. *Nature* 568 (2019), 493–498.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein GAN. arXiv:1701.07875 [stat.ML]

[4] Edward F Chang. 2020. Human ECoG speaking consonant-vowel syllables. https://figshare.com/collections/Human_ECoG_speaking_consonant-vowel_syllables/4617263

[5] Isaac A. Corley and Yufei Huang. 2018. Deep EEG super-resolution: Upsampling EEG spatial resolution with Generative Adversarial Networks. In *2018 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*. 100–103. https://doi.org/10.1109/BHI.2018.8333379

[6] Fatemeh Fahimi, Zhuo Zhang, Wooi Boon Goh, Kai Keng Ang, and Cuntai Guan. 2019. Towards EEG Generation Using GANs for BCI Applications. In *2019 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*. 1–4. https://doi.org/10.1109/BHI.2019.8834503

[7] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. arXiv:1406.2661 [stat.ML]

[8] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. Improved Training of Wasserstein GANs. arXiv:1704.00028 [cs.LG]

[9] Kay Gregor Hartmann, Robin Tibor Schirrmeister, and Tonio Ball. 2018. EEG-GAN: Generative adversarial networks for electroencephalograhic (EEG) brain signals. *ArXiv* abs/1806.01875 (2018).

[10] Christian Herff, Dominic Heger, Adriana De Pesters, Dominic Telaar, Peter Brunner, Gerwin Schalk, and Tanja Schultz. 2015. Brain-to-text: Decoding spoken phrases from phone representations in the brain. *Frontiers in Neuroscience* 9 (06 2015). https://doi.org/10.3389/fnins.2015.00217

[11] Tian jian Luo, Ya chao Fan, Lifei Chen, Gongde Guo, and Changle Zhou. 2020. EEG Signal Reconstruction Using a Generative Adversarial Network With Wasserstein Distance and Temporal-Spatial-Frequency Loss. *Frontiers in Neuroinformatics* 14 (2020).

[12] Isaak Kavasidis, Simone Palazzo, Concetto Spampinato, D. Giordano, and Mubarak Shah. 2017. Brain2Image: Converting Brain Signals into Images. 1809–1817. https://doi.org/10.1145/3123266.3127907

[13] Yi Kwang Moo, Trulls Eduard, Ono Yuki, Lepetit Vincent, Salzmann Mathieu, and Pascal Fua. 2018. Learning to Find Good Correspondences. *CVPR* (2018).

[14] Jesse Livezey, Kristofer Bouchard, and Edward Chang. 2018. Deep learning as a tool for neural data analysis: Speech classification and cross-frequency coupling in human sensorimotor cortex. *PLOS Computational Biology* 15 (03 2018). https://doi.org/10.1371/journal.pcbi.1007091

[15] David Moses, Nima Mesgarani, Matthew Leonard, and Edward Chang. 2016. Neural speech recognition: Continuous phoneme decoding using spatiotemporal representations of human cortical activity. *Journal of Neural Engineering* 13 (10 2016), 056004. https://doi.org/10.1088/1741-2560/13/5/056004

[16] Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *CoRR* abs/1511.06434 (2016).

[17] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved Techniques for Training GANs. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (Barcelona, Spain) *(NIPS'16)*. Curran Associates Inc., Red Hook, NY, USA, 2234–2242.

[18] Tanja Schultz, Michael Wand, Thomas Hueber, Dean J. Krusienski, Christian Herff, and Jonathan S. Brumberg. 2017. Biosignal-Based Spoken Communication: A Survey. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25, 12 (2017), 2257–2271. https://doi.org/10.1109/TASLP.2017.2752365

[19] Sim Swee, Desmond Teck Kiang Kho, and Lim You. 2016. EEG Controlled Wheelchair. *MATEC Web of Conferences* 51 (01 2016), 02011. https://doi.org/10.1051/matecconf/20165102011

[20] Sun Weiwei, Jiang Wei, Trulls Eduard, Tagliasacchi Andrea, and Yi Kwang Moo. 2018. ACNe: Attentive Context Normalization for Robust Permutation-Equivariant Learning. *CVPR* (2018).