GAN-based Electroencephalograhic (EEG) Brain Signals Generation

Wenyuan Zhao, Lindong Ye and Ziyan Cui
Department of Electrical and Computer Engineering
University of California San Diego
La Jolla, CA 92093
{wez030, lye, z5cui}@ucsd.edu

Abstract

Processing and analysis of brain signals generally requires a large amount of data. But the acquisition of EEG signals is difficult while the sample size of the data set is small, and sometimes categories are unbalanced in the data set. Based on the challenge, we proposed the WGAN-GP method, a variant of GAN, to generate and extend the data set of EEG signals. Our experiments on both single-channel and multi-channel model show that the generated EEG signals have closer shape and better spectrum performance than those generated by traditional methods. Our results and analysis show that the proposed WGAN-GP can generate accurate and diverse EEG signals, and thus, help extend the data set which is difficult to collect physically. We've made the code associated with this work available at https://github.com/warrenzha/GAN-EEG-generation.

1 Introduction

An important research direction in brain science is the analysis and processing of brain signals. Brain-computer interface systems (BCIs) can connect the brain with external world by extracting features of electroencephalograhic (EEG) brain signals. Scientists can obtain desired information via signal acquisition, pre-processing and feature extraction. However, it is difficult to collect EEG signals from subjects by repeating the task for a long time, and it is even more difficult for the elderly and children, who are the more significant subjects of brain research. In addition, most of the current BCIs algorithms are working offline on collected data sets. For real-time BCIs systems, the limited data set will affect the speed of real-time training and decision making procedure. Therefore, it is necessary to expand the data set by generating new EEG signals based on generative models.

With the development of deep learning models, more and more deep learning models and algorithms are applied to the classification and feature extraction of Electroencephalography (EEG) signals. However, the downside exist as the acquisition of EEG signals is difficult while the sample size of the data set is small, and sometimes categories are unbalanced. In this way, generating EEG signals from generative models is an relatively effective alternative, as it augments the data by producing artificial samples not included in the original data set and thereby increase training data to improve the model. Our purpose is to design a generative model to generate useful EEG signals for BCIs research.

Contribution of this project includes:

- Developed the network structure for EEG signal generation, according to single-channel (1D-convolution, 1D-deconvolution, 1D-pooling layer) and multi-channel information (2D-convolution, 2D-deconvolution and 2D-pooling layer). Implemented the generation experiment on P300 EEG signals.

Preprint. Under review.

- Visualized the generated data from two perspectives, frequency and time domains, and evaluate model stability in training. The experimental results show that our generative model, WGAN-GP, is stable, and the generated signal is accurate and diverse.
- Compared our model with traditional generative model, VAE. The spectrum performance shows that our model performs better than VAE. The signals generated by WGAN-GP improve classification models more than signals generated by VAE.

2 Related Works

Model-based methods for signals generation. Before the advent of GAN, the ways to generate new signals are generally model-based. Li Y et al. and Wang F et al. generated new brain signals by adding Gaussian noise to the signals [7]. Majidov I et al. generated new brain signals using sliding window [10]. Shovon T H. et al. generated new brain signals by performing short-time Fourier transform(STFT) on signals, then performing a geometric transformation on the generated pictures [13]. Lotte F. generated new brain signals in 2011 by a split recombination method that considered only time domain features [8]. In 2015, they further improved their previously proposed method by proposing a split and recombination method that takes into account both time domain characteristics and frequency domain characteristics [9].

Data-driven methods for signals generation. Proposed by Goodfellow I J in 2014, generative Adversarial Networks (GAN) is first widely used in image generation [5]. In the early days of GAN, however, it was not used in the field of time series signaling, especially in brain signal generation. It suffered from training instability and were restricted to low resolution images. A lot of advancement in regard to stability and the quality of the generated images has been made with different regularization methods [2]. GANs also manipulate properties in generated samples [12] and therefore can help understand the original data distribution in GAN training session. Until recent years, researchers started to generate EEG signals based on GAN methods. The variants of GAN have been improved from the original GAN for generation of EEG signals, especially Motor Imagery (MI) signals. Esteban C et al. used Recurrent GAN (RGAN) and Recurrent Conditional GAN (RCGAN) to generate medical time series signals [4]. Abdelfattah S M et al. used RGAN to generate motion imaging signals in 2018 [1]. Hartmann K G et al., in 2018, used improved Wasserstein GAN (WGAN) to generate single-channel EEG signals, and demonstrated through a series of indicators that they can generate natural and artificial signals that approximate the original signal [6]. Panwar S et al. generated RSVP single-channel signals in 2019, using gradient penalty-based conditional WGAN [11]. Aznan N K N et al. used Deep Convolutional GAN (DCGAN), WGAN, and VAE to generate single-object and multi-object Steady-State Visual Evoked Potential (SSVEP) signals, respectively [3].

P300 brain signal. The brain signal stuied in this project is the P300 signal, which is a type of Event-related Potentials (ERP) that can be visually induced. ERP has many components, which can be divided into two parts - endogenous components (such as N200, P300, N400) and exogenous components (such as P80, N100, P200). P300 is the most stable one and has received a lot of attention in research. Although there have been some studies on EEG signal generation based on GAN variants, most of them focus on motor imagery signals and other kinds. Few works studied visual evoked-related signals. Through literature review, no research generated P300 signals based on GAN methods.

3 Problem Definition

The purpose of this project is to generate useful EEG signals (P300) to help expand the data set for BCIs research. As shown in Figure 1, P300 refers to a positive spike generated in the area near the top of the head about 300ms after stimulation.

The structure of the brain is unique, and through the same experiments, the shape of P300 waveform from different subjects will be quite different. Figure 2 shows the comparison of the average P300 signals between two different subjects. It shows significant differences in the peak value and the position of the wave peak.

P300 signals studied in this project were Wadsworth BCI Dataset (P300 Evoked Potentials) recorded a complete set of P300 evoked potentials with BCI2000 by Donchin et al., 2000. The recorded data

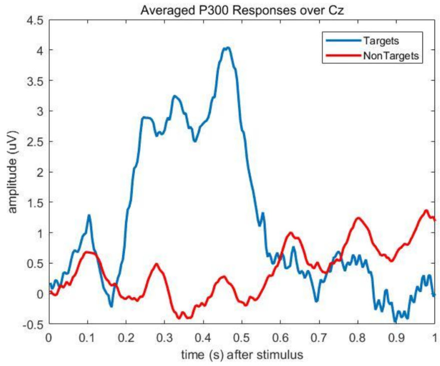

Figure 1: P300 waveform (blue curve)

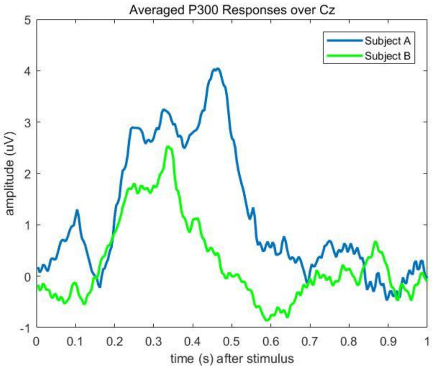

Figure 2: P300 signal from 2 subjects

Table 1: Obtained data size and data pattern.

	Positive sample size	Negative sample size	Ratio	Matrix size of each sample
Subject A	170	850	1:5	[8,240]
Subject B	170	850	1:5	[8,240]

has been converted into 4 Matlab *.mat files, one training (85 characters) and one test (100 characters) for each of the two subjects A and B. The data set was obtained by using the P300 character spelling system using the row-column stimulation paradigm and using the 64-channel EEG acquisition method in the international 10-20 electrode specification. Figure 3 shows the electrode names and channel assignment numbers for the 64 channels. For each *.mat file, the recorded 64-channel EEG signal is organized in one big matrix.

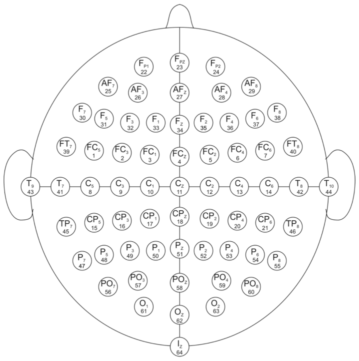

Figure 3: This diagram illustrates electrode designations (Sharbrough, 1991) and channel assignment numbers as used in our experiments.

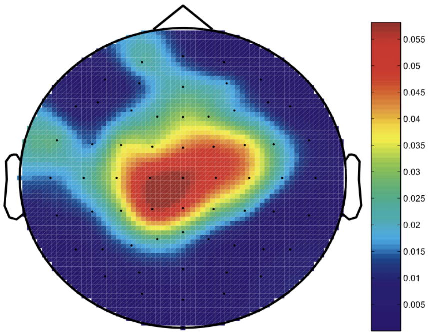

Figure 4: This figure shows a topography of values of P300 from different channels. Warmer color means more active brain signal.

During signal acquisition, the EEG signal has been filtered by a band-pass filter from 0.1-60Hz and digitized under 240Hz. Figure 4 indirectly indicates spatial differences in P300 stimulation, with stronger P300 signal excitation near the top of the brain. Therefore, in this project, 8 channels with high variance in Figure 4 are selected for the generation of EEG signals. Then the obtained data size and number of samples to deal with are shown in Table 1.

Due to the difference in the number of targets and non-targets, there are obvious differences in the number of positive and negative samples. In this project, we work on generating target P300 signals expand positive samples and eliminate the imbalance of sample categories.

4 Methodology

4.1 Generative adversarial network (GAN)

Native GAN was proposed by Goodfellow I J in 2014 [5], which is an unsupervised architecture consisting of a generator network (G) and a discriminator network (D) shown in Figure 5. The inputs of G and D are are real data x and random variable z, respectively. Then $G(z)$ is the generated data generated by G that obeys the real data distribution p_{data} as much as possible. The discriminator D is to complete the binary classification task of the input data, and correctly divide the data into two categories: true (real data x) and false (generated data $G(z)$). Our goal is to make the performance $D(G(z))$ and $D(x)$ as equal as possible. During this process of adversarial and iterative optimization, D and G are continuously improved. When D cannot correctly identify the data source, it can be considered that generator G has learnt the distribution of the real data, i.e. $p_{data}(x) = p_g(x)$.

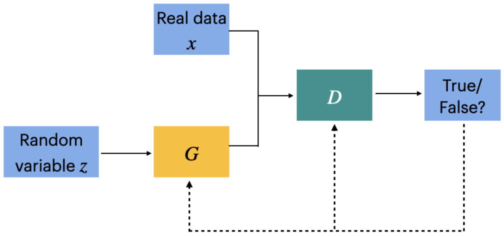

Figure 5: The frame of GAN.

The training process of GAN is shown in Algorithm 1. First, fix G and optimize D. Since the last layer of D is a *Sigmoid* function layer, the output of D is limited to $[0, 1]$, which also represents the probability of the authenticity of the data. The process of training the discriminator is equivalent to minimizing the cross-entropy loss function:

$$Loss_D(\theta_D, \theta_G) = -\frac{1}{2}\left(E_{x\sim p_{data}(x)}\left[\log D(x)\right] + E_{z\sim p_{data}(z)}\left[\log\left(1 - D\left(G\left(z\right)\right)\right)\right]\right). \quad (1)$$

Then, fix D and optimize G. When the input is $G(z)$, the discriminator is to discriminate it as false. And the generator try to fool the discriminator not to judge it as false, but to judge it as true. When the accuracy of D is minimized, the generator G reaches the global optimal solution $p_{data}(x) = p_g(x)$. Then the objective of GAN optimization can be given by

$$\min_G \max_D L(D, G) = E_{x\sim p_{data}(x)}\left[\log D(x)\right] + E_{z\sim p_{data}(z)}\left[\log\left(1 - D\left(G\left(z\right)\right)\right)\right] \quad (2)$$

Algorithm 1: GAN mini-batch stochastic gradient descent

```
for training episode do
    for Discriminator episode do
        Sample a minibatch of $m$ noisy data with distribution $p_g(z)$: $\{z_1, z_2, \ldots, z_m\}$
        Sample a minibatch of $m$ real data with distribution $p_{data}(x)$: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$
        Update discriminator with gradient ascent:
            $\nabla\theta_d \frac{1}{m}\sum_{i=1}^{m}\left[\log D(x^{(i)}) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)\right]$
    end
    Sample a minibatch of $m$ noisy data with distribution $p_g(z)$: $\{z_1, z_2, \ldots, z_m\}$
    Update generator with gradient descent:
        $\nabla\theta_g \frac{1}{m}\sum_{i=1}^{m}\left[\log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)\right]$
end
```

4.2 Wasserstein GAN (WGAN)

One big drawback of GANs is the notorious instability of the discriminator during training. The discriminator might collapse into only recognizing few and narrow modes of the input distribution as real, which drives the generator to produce only a limited amount of different outputs. The original GAN framework tries to minimize the Jensen-Shannon (JS) divergence between the real data distribution P_{data} and generated data distribution P_g. If the discriminator is too strong, it will easily separate the generated data from the real data, resulting in the inability to provide a sufficiently large gradient for the generator to update, resulting in the problem of vanishing gradients for the generator.

Wasserstein GANs (WGAN) and their improved version proposed by Arjovsky et al. (2017) [2] show promising advances for training stability. They proposed to to minimize the *Wasserstein* distance between the distributions instead of the JS-divergence. This leads the discriminator to optimize the Wasserstein difference

$$W(P_r, P_g) = \frac{1}{K} \sup_{\|D\|_L \leqslant K} E_{x_r \sim P_{data}} \left[D(x_r) \right] - E_{x_g \sim P_g} \left[D(x_g) \right], \tag{3}$$

where the constraint is the Lipschitz continuity condition. That is, there exists a constant $K \geqslant 0$, such that

$$\left| f(x_1) - f(x_2) \right| \leqslant K \left| x_1 - x_2 \right|, \ \forall x_1, x_2 \in dom f.$$

4.3 GAN improvement: WGAN-GP

During training procedure, to maximize the loss for the discriminator, one explicit way is to increase the difference between score of the real sample and the generated sample. For the discriminator, the optimal strategy is to make each parameter either take the upper bound or the lower bound, causing the parameter concentration problem.

Hence, we add a gradient penalty term to the loss function of the discriminator to solve this problem. The Lipschitz condition requires that the gradient of the discriminator does not exceed K, which is the role of the gradient penalty term. Then the loss function of the discriminator is given by

$$Loss_D = E_{x_r \sim P_{data}} \left[D(x_r) \right] - E_{x_g \sim P_g} \left[D(x_g) \right] - \lambda E_{x \sim P_{penalty}} \left[\max \left(0, \|\nabla_x D(x)\| - 1 \right) \right]. \tag{4}$$

If the gradient penalty coefficient is chosen higher, the gradient of the discriminator can be limited around 1, which satisfies the 1-Lipschitz condition well, and stabilizes the gradient. The improved GAN is shown in Algorithm 2.

Algorithm 2: WGAN-GP

```
Input: $\lambda$: gradient penalty coefficient
Initialization: $\omega_0$: initial discriminator parameters; $\theta_0$: initial generator parameters;
while $\omega$ does not converge do
    for $t = 1, \ldots, n$ do
        for $i = 1, \ldots, m$ do
            Sample a minibatch of $m$ real data $\left\{x^{(i)}\right\}_{i=1}^{m} \sim P_{data}$;
            Sample a minibatch of $m$ noisy data $\left\{z^{(i)}\right\}_{i=1}^{m} \sim P_z$;
            $\tilde{x} \leftarrow G_\theta(z)$;
            $\hat{x} \leftarrow \epsilon x + (1 - \epsilon)\tilde{x}$;
            $L^{(i)} \leftarrow D(\tilde{x}) - D(x) + \lambda \left( \|\nabla_{\hat{x}} D\left(\hat{x}\right)\|_2 - 1 \right)^2$;
        end
        $w \leftarrow Adam\left(\nabla_\omega \frac{1}{m} \sum_{i=1}^{m} L^{(i)}, \omega, \alpha, \beta_1, \beta_2\right)$;
    end
    Sample a minibatch of $m$ random data from a given distribution: $\left\{z^{(i)}\right\}_{i=1}^{m} \sim P_z$;
    $\theta \leftarrow Adam\left(\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} L^{(i)} - D(G_\theta(z)), \theta, \alpha, \beta_1, \beta_2\right)$;
end
```

Table 2: Network Topology of Generator

Generator input: [1,100] under distribution N(0,1)				
Network Layers	Input Size	Output Size	# Parameters	Hyperpara.
Affine Layer	[1,100]	[1,400]	40400	-
Leaky-ReLU	[1,400]	[1,400]	1	α
Reshape	[1,400]	[16,25]	0	-
ConvTranspose1d	[16,25]	[16,56]	2064	k, s
Leaky-ReLU	[16,25]	[16,56]	1	α
ConvTranspose1d	[16,25]	[16,118]	2064	k, s
Leaky-ReLU	[16,118]	[16,118]	1	α
ConvTranspose1d	[16,25]	[8,240]	776	k, s
Total # parameters: 45307				

Table 3: Network Topology of Discriminator

Discriminator input: [8,240]				
Network Layers	Input Size	Output Size	# Parameters	Hyperpara.
Conv1d	[8,240]	[16,118]	784	k, s
Leaky-ReLU	[16,118]	[16,118]	1	α
Conv1d	[16,118]	[16,56]	2064	k, s
Leaky-ReLU	[16,56]	[16,56]	1	α
Max-pool	[16,56]	[16,28]	0	k
Reshape	[16,28]	[1,448]	0	-
Affine Layer	[1,448]	[1,100]	44900	-
Leaky-ReLU	[1,100]	[1,100]	1	α
Affine Layer	[1,100]	[1,1]	101	-
Total # parameters: 47852				

5 Experiment and Results

5.1 Data pre-processing

Although a 0.1–60Hz band-pass filter has been carried out when the signal is collected, there still exists high frequency noise. While main frequency of P300 is within 20Hz, the noise still has a negative effect on the following analysis. Therefore, applying low-pass or band-pass filter to eliminate the high frequency noise is necessary.

In this project, we feed the original signal into a Butterworth low-pass filter to filter the high frequency noise. The sampling rate f_s = 240Hz, pass-bands f_{pass} = 15Hz, stop-bands f_{stop} = 20Hz. The main frequency part of P300 (within 20Hz) are maintained.

5.2 Network topology and experimental parameters

In this work, we generated P300 EEG signals with WGAN-GP method from both single-channel and multi-channel. From Section 3, the P300 signal was sampled for 1 second under 240Hz digital frequency from 64 channels. Hence, each sample is a matrix with the size $[240, 64]$, of which each column vector represents the sampled P300 signal of a specific channel.

For single-channel implementation, the input of the discriminator is a vector signal from a specific channel. For multi-channel implementation, we selected 8 channels with highest variance for the generation. Hence, the input becomes a matrix with the size $[8, 240]$. Both the generator G and discriminator D are trained by convolution neural network (CNN). The details of the network topology are shown in Table 2 and 3.

The generator and discriminator are trained by Adam optimizer. The learning rate of the discriminator $lr_1 = 0.00005$, and the learning rate of the generator is $lr_2 = 0.0001$. Setting the learning rate of the discriminator to be slightly smaller than the learning rate of the generator is mainly to make the discriminator not too powerful, which will affect the stability of training and the learning speed of the generator. The two parameters of Adam optimizer $\beta_1 = 0.5, \beta_2 = 0.99$.

The total epochs of training is set to 2000. In each training epoch, train the discriminator for 5 times and then train the generator once. The size of mini-batch is set to 10. By packing the input model in small batches instead of individual data-input models, we can speed up the training.

5.3 Generated data visualization

The most intuitive way to observe the signal is the signal waveform in time domain. We directly plot the waveform of generated P300 signals in Figure 6. Our experiment generated a total of 800 signal samples, from which 170 samples (the size of the real data set) were taken for visualization. The left column of Figure 6 is the generated signal in each channel, while the right column of Figure 6 is the real signal in each channel. The ordinate is the normalized amplitude of the signal, and the abscissa is time.

From an overall amplitude perspective, the amplitude of the generated signal is not much different from the real signal. In the real data, there are individual signals that are outliers, and their amplitudes are quite different from most of the signals, but in the generated data, no outlier signals that are far from most of the signals are found, which shows that the training of the network is still good. Better, avoid the influence of outlier samples and learn the main characteristics of the distribution.

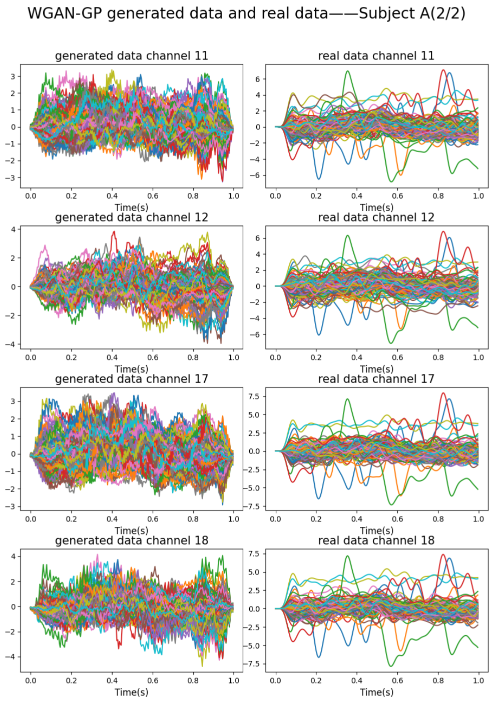

Figure 6: WGAN-GP generated P300 signal in 8 channels.

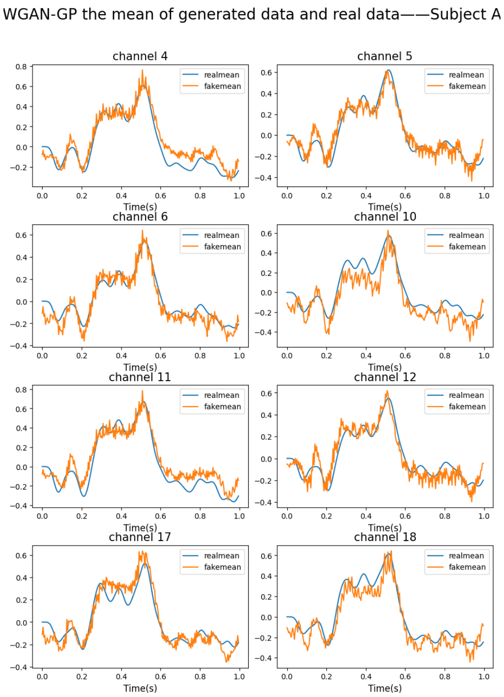

Figure 7: Comparison: WGAN-GP generated signal and real signal.

Figure 7 shows the comparison between generated signals and real signals in each channel. Although the generated signal is not as smooth as real signal, this does not affect the qualify of our generated signals since the raw data are filtered by a low-pass filter. Hence, the raw signals should look more smooth than our generated signals. The peak at around 0.5s illustrates that the network learns the property of P300. Comparing signals generated by different channels, each signal is unique and has some differences from real signals, which means single-channel signal does not copy the real signal or each other. Moreover, the means of generated signals and real signals are the same. Therefore, our generated signals can help expand the data set based on the results in time domain.

5.4 Spectrum performance compared with VAE

Frequency and spectrum are important indicator characteristics of a signal. For generating a signal, the observation of the spectrum is essential in order to evaluate the quality of its generation. For the P300 EEG signal, since its main component is within 20Hz, the spectrum of the generated signal mainly observes two points: the change of the spectrum within 20Hz compared with the real signal spectrum, and whether there is high frequency in the high frequency band outside 20Hz noise introduction.

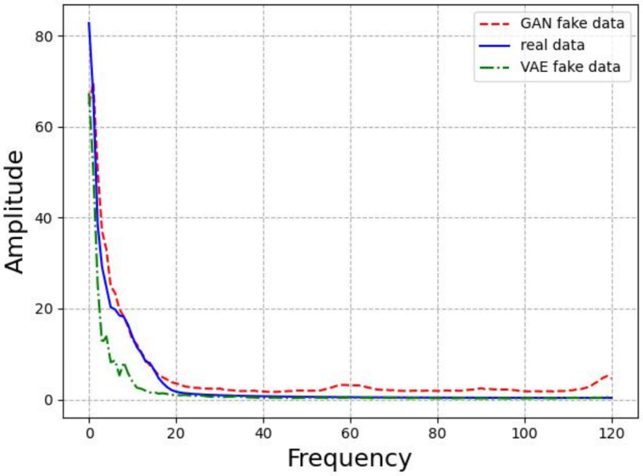

Figure 8: Spectrum performance from WGAN-GP, VAE, and raw data set.

We compared the spectrum performance between P300 signals from our WGAN-GP model, traditional VAE model, and raw data set. Figure 8 is the frequency domain comparison of generative models with Subject A as an input. In high frequency domain, VAE behaves better than WGAN-GP, because it does not have high frequency noise. However, in the low frequency domain, especially main domain, WGAN-GP apparently has better performance, and simulates the feature of real signals well. Our focus of observation should be the difference between real signal and generated signal within 20Hz since the main frequency of P300 is within 20Hz. The high frequency noise can be eliminated easily by a low-pass filter.

5.5 WGAN-GP model evaluation

As mentioned Section 4, WGAN-GP relies on minimizing the loss functions of the generator and discriminator. By observing the convergence of the loss function, that is, observing the training situation, the WGAN-GP modelcan be evaluated, or the generated signal can be indirectly evaluated by observing whether there is overfitting.

Figure 9 shows how the loss function converged in our experiment. The Wasserstein distance of two distribution can be approximated by the inverse number of loss function in WGAN. The loss function of discrinimator (Dloss), which is positive at the beginning, drops significantly, fluctuates until it converges to a negative number close to zero. This illustrates that the discriminator cannot discriminate the feature of generated signal from real signal, that is, generator has learnt the characteristics of real P300 signal.

The loss function of generator (Gloss) is arguably meaningless. But according to the essence of the adversarial network, the loss function of the generator should be in a fluctuating state, which reveals "adversary". Hence, the training process of WGAN-GP generative model is convergent.

We then selected four evaluation parameters that are widely used in GAN evaluation to compare the performance of our WGAN-GP model with traditional generative model VAE in Table 4:

- **Inception score (IS).** Feed the generated samples into the network, and predict the labels. Calculate the entropy of the conditional label distribution of the generated samples to

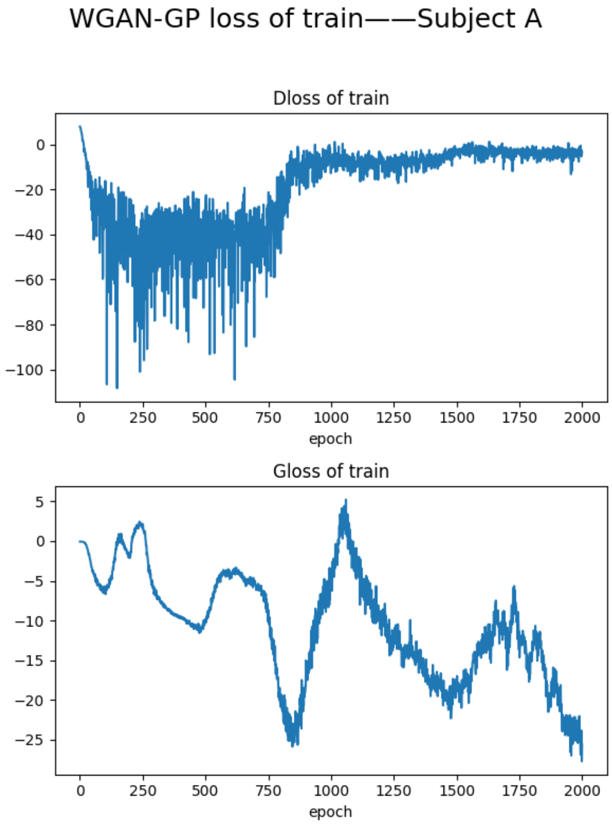

Figure 9: Convergence rate of WGAN-GP.

Table 4: Evaluation of WGAN-GP and VAE.

	IS	FID	SWD	MMD
WGAN-GP	1.3220	**22.6670**	**0.022**	**0.0726**
VAE	**1.4421**	32.2682	0.0232	0.0524

measure the quality of the samples. When the value of IS is higher, it indicates that the quality and diversity of the generated samples are better.

- **Frechet Inception score (FID)** FID evaluate the model by measuring the distance between the generated sample distribution and the true sample distribution. With lower FID, the two sample distributions are closer, indicating better quality and diversity.

- **Sliced Wasserstein distance (SWD).** SWD maps to the Wassertein distance between two distributions. If the SWD is lower, the two distributions are more similar in appearance and diversity.

- **Minimum average Manhattan Distance (MMD).** MD calculates the distance between real samples and generated samples to evaluate similarity.

When evaluated by IS, VAE performs better since it generates less high frequency noise. However, when evaluated by FID, WGAN-GP shows more diversity, thus, WGAN-GP is better than VAE. In terms of SWD, two models are nearly the same rating. Generally, WGAN-GP can generate accurate and more diverse P300 signals.

6 Future Works

There are still many open possibilities for further investigations. First, the data set is a sample of average elimination effects and is small in size. We have to determine whether the model can be further used in a large sample. Second, in practice, most BCIs researches use full-channel EEG signals. The WGAN-GP generative model in multi-channel, however, played with only 8 selected channels. We would like to extend WGAN-GP to full channel. Third, due to the imbalance of categories, the need for sample supplementation is relatively usual. It is necessary to further improve the model performance through adjustment for practical real-time classification applications.

7 Conclusion

This project focuses on GAN-based EEG signal generation algorithms. We generated useful P300 EEG signals based on an improved WGAN-GP model. Experimental results show that WGAN-GP generative model is stable and accurate under single and multi-channel. Model evaluations show that WGAN-GP has better spectrum performance than traditional method VAE. WGAN-GP also performs better than VAE in terms of signal diversity, and WGAN-GP improves the classification models more than VAE.

References

[1] Sherif M Abdelfattah, Ghodai M Abdelrahman, and Min Wang. Augmenting the size of eeg datasets using generative adversarial networks. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2018.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017.

[3] Nik Khadijah Nik Aznan, Amir Atapour-Abarghouei, Stephen Bonner, Jason D Connolly, Noura Al Moubayed, and Toby P Breckon. Simulating brain signals: Creating synthetic eeg data via neural-based generative models for improved ssvep classification. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[4] Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*, 2017.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[6] Kay Gregor Hartmann, Robin Tibor Schirrmeister, and Tonio Ball. Eeg-gan: Generative adversarial networks for electroencephalograhic (eeg) brain signals. *arXiv preprint arXiv:1806.01875*, 2018.

[7] Yang Li, Xian-Rui Zhang, Bin Zhang, Meng-Ying Lei, Wei-Gang Cui, and Yu-Zhu Guo. A channel-projection mixed-scale convolutional neural network for motor imagery eeg decoding. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(6):1170–1180, 2019.

[8] Fabien Lotte. Generating artificial eeg signals to reduce bci calibration time. In *5th International Brain-Computer Interface Workshop*, pages 176–179, 2011.

[9] Fabien Lotte. Signal processing approaches to minimize or suppress calibration time in oscillatory activity-based brain–computer interfaces. *Proceedings of the IEEE*, 103(6):871–890, 2015.

[10] Ikhtiyor Majidov and Taegkeun Whangbo. Efficient classification of motor imagery electroencephalography signals using deep learning methods. *Sensors*, 19(7):1736, 2019.

[11] Sharaj Panwar, Paul Rad, John Quarles, and Yufei Huang. Generating eeg signals of an rsvp experiment by a class conditioned wasserstein generative adversarial network. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 1304–1310. IEEE, 2019.

[12] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[13] Tanvir Hasan Shovon, Zabir Al Nazi, Shovon Dash, and Md Foisal Hossain. Classification of motor imagery eeg signals with multi-input convolutional neural network by augmenting stft. In *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*, pages 398–403. IEEE, 2019.