# Data Augmentation for P300-based Brain-Computer Interfaces Using Generative Adversarial Networks

Kassymzhomart Kunanbayev* Berdakh Abibullaev† Amin Zollanvari*

*Electrical and Computer Engineering Department, School of Engineering and Digital Sciences, Nazarbayev University, Kazakhstan
†Robotics and Mechatronics Department, School of Engineering and Digital Sciences, Nazarbayev University, Kazakhstan
Emails: {kassymzhomart.kunanbayev, berdakh.abibullaev, amin.zollanvari}@nu.edu.kz

2021 9th International Winter Conference on Brain-Computer Interface (BCI) | 978-1-7281-8485-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/BCI51272.2021.9385317

***Abstract*—A notable problem in Brain-Computer Interface (BCI) is the burden of collecting an adequate amount of training data to be used in estimating a robust classifier model. This study explores the prospect of using Generative Adversarial Networks (GANs), a novel family of generative models, to perform data augmentation for generating artificial training data that are used in the classification of P300 event-related potentials in electroencephalogram (EEG) signals. In this regard, we consider two popular GANs, namely, Deep Convolutional Generative Adversarial Networks (DCGAN) and Wasserstein GAN with Gradient Penalty (WGAN-GP), and explore their efficacy in generating artificial EEG data in both subject-specific and subject-independent contexts. The results show that data augmentation using both DCGAN and WGAN-GP could lead to improved classification performance. However, the operating conditions that an improvement is observed differ between the subject-specific and subject-independent classification schemes. In particular, we observe that a transition from a relatively small to large training sample size per subject would generally lead to a better classification performance in training subject-specific classifiers; however, when a limited number of subjects is available, this transition could potentially result in an opposite effect in case of subject-independent classification.**

***Index Terms*—generative adversatial networks, P300, data augmentation, subject-specific, subject-independent**

## I. Introduction

BRAIN-computer interfaces (BCI) based on Event-Related Potentials (ERP) elicited in electroencephalogram (EEG) represent an important class of neural interface systems widely used for communication and control [1], [2].

A prominent ERP component is the P300 waveform, which is a positive wave evoked 300-400 milliseconds after the stimulus onset when a user mentally attends to a rare target sensory event. Analyzing P300 waves plays a key role in designing BCI spellers that have been used as important means of communication for patients suffering from motor neuron diseases. As these patients need a communication channel without involving muscular movement, monitoring brain signals remains crucial. In this regard, P300-based BCIs offer non-invasive, relatively cheap, mobile, and easy-to-set-up tools compared to others [3].

However, despite the attainable cost and availability, the collection and decoding of P300 waves retain certain challenges. P300 waves tend to be high-dimensional, affected by noise and include poor signal-to-noise-ratio [4], [5]. Apart from that, high inter-subject variability is another challenge because of the non-stationary of the EEG signals, individual psychological and physiological factors of subjects, and measurement settings [6], [7]. All these issues render the process of collecting P300s laborious.

In practice it is common to face inadequate sample size that often leads to problems such as class imbalance, insufficient diversity, or insufficient amount of data that is used in training classifiers of P300 waves. An approach that has been put forward is the data augmentation using which we can generate new artificial data by transforming the present data. This could potentially lead to a significant reduction in the time required to calibrate the BCI. In this regard, a number of conventional data augmentation techniques have been already used: signal segmentation and recombination, and artificial trail generation [8], [9], rotational data augmentation [10], upsampling minority class, change in morphology, dataset-to-dataset transfer [9]; label-based data augmentation [11]. Insofar as P300 waves are concerned, Krell *et al.* [12] proposed several data augmentation techniques using temporal, rotational distortions, and spatial approaches that led to classification performance improvement in the range of 1% and 6%.

Deep learning methods have also been broadly used to augment EEG data. Naturally, and not surprisingly, a number of studies have examined applications of deep generative models to generate artificial EEG data. The two major representatives of generative models are Variational Autoencoders (VAE) [13] and Generative Adversarial Networks (GAN) [14]. In particular, GANs have been actively investigated in various applications such as images and videos to generate artificial data [15], [16]. In BCI, conditional deep convolutional GAN (cDCGAN) has been used to synthesize artificial EEG data after transforming real data to the time-frequency domain [17]. In [18], data augmentation using cDCGAN was used to show an improved leave-one-subject-out classification accuracy in both focused and diverted attention motor imagery tasks.

This work was partially supported by the Faculty Development Competitive Research Grants Program of Nazarbayev University under grant numbers 021220FD2051, 021220FD1151, and NPO Young Researchers Alliance and Nazarbayev University Corporate Fund "Social Development Fund" for grant under their Fostering Research and Innovation Potential Program

In [19], the creation of artificial steady-state visual evoked potential-based EEG data using different generative models is investigated. Panwar *et al.* [20] introduced an altered version of gradient penalty-based Wasserstein GANs (WGAN-GP) with class conditioning to classify rapid serial visual presentation (RSVP) ERPs. In this work, we study the efficacy of both DCGAN and WGAN-GP in generating artificial P300 waves in both subject-specific and the subject-independent classification schemes.

## II. Materials and Methods

### A. Dataset description

In this study, we utilized the dataset collected by P. Arico *et al.* [21] from a P300-based BCI Speller system used an overt attention mode (the data is accessible via the MOABB system [22]). There were 10 healthy subjects who participated in the study (female, mean age 26.8 ± 5.6). The subjects were instructed to concentrate on 36 distinct characters arranged in a Farwell and Donchin BCI speller and to attend one of the characters during the given character selection epoch. 16 Ag/AgCl electrodes (Fz, FCz, Cz, CPz, Pz, Oz, F3, F4, C3, C4, CP3, CP4, P3, P4, PO7, PO8) have been placed according to the 10-10 standard and were referenced to the earlobes and grounded to the right mastoid. The EEG signals were collected using g.USBamp amplifier with a sampling rate of 256 Hz and were filtered using high-pass and low-pass filters with cutting frequencies of 0.1 and 20 Hz, respectively.

### B. Generative Adversarial Networks

First introduced by Goodfellow *et al.* [14], the idea of generative adversarial networks (GAN) is based on a two-player minimax game, in which there are two models: a generator $G$ that aims to fool the discriminator $D$ by producing plausible samples $\boldsymbol{x}$ from the noise distribution $p_{\boldsymbol{z}}(\boldsymbol{z})$, and a discriminator $D$ that tries to distinguish the samples from the generator's distribution and the data distribution $p_{data}$. This can be shown in terms of loss function $\mathsf{L}(G, D)$

$$\min_G \max_D \mathsf{L}(G, D) = \mathrm{E}_{\boldsymbol{x}\sim p_{data}}[log D(\boldsymbol{x})] + \mathrm{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[log(1 - D(G(\boldsymbol{z})))] \tag{1}$$

There are plenty of works demonstrating successful implementations of GAN in generating samples of the image, video, and text [15], [16], [23]. Besides, various improvements and variations to GANs have been proposed that leverage the quality of the generated data [24]–[27]. In this study, we investigate the two most popular GANs to augment the P300 data: deep convolutional GAN (DCGAN) and Wasserstein GAN with gradient penalty (WGAN-GP).

*1) DCGAN:* Radford *et al.* [24] first proposed a generative model based on using deep convolutional layers and recommended a set of improvements in building a model. Generally, DCGAN is implemented based on the loss function in Eq. 1. In this article, we follow the recommendations provided by the authors and implement DCGAN as follows. The generator model consists of two 2D transposed convolution layers each

[여기에 입력]

followed by batch normalization and ReLU operators and one 2D convolutional layer followed by the *tanh* activation function. The discriminator model includes three 2D convolutional blocks, where each includes one 2D convolutional layer with Leaky ReLU activation. The output of the last convolutional block is flattened into a vector to be processed into a fully-connected layer followed by a sigmoid activation.

*2) WGAN-GP:* The original implementation of GAN has certain problems that lead to the failure of the training such as vanishing gradients and mode collapse, as described by [28]. However, these issues were addressed by Arjovsky *et al.* [26] by incorporating a different loss function based on Wasserstein distance, and it was further improved by incorporating a gradient penalty term in the loss function (1) as follows [29]:

$$\min_G \max_D \mathsf{L}(G, D) = \mathrm{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[D(G(\boldsymbol{z}))] - \mathrm{E}_{\boldsymbol{x}\sim p_{data}}[D(\boldsymbol{x})] + \lambda\, \mathrm{E}_{\hat{\boldsymbol{x}}\sim \mathsf{P}_{\hat{\boldsymbol{x}}}}[(||\nabla D(\hat{\boldsymbol{x}})||_2 - 1)^2] \tag{2}$$

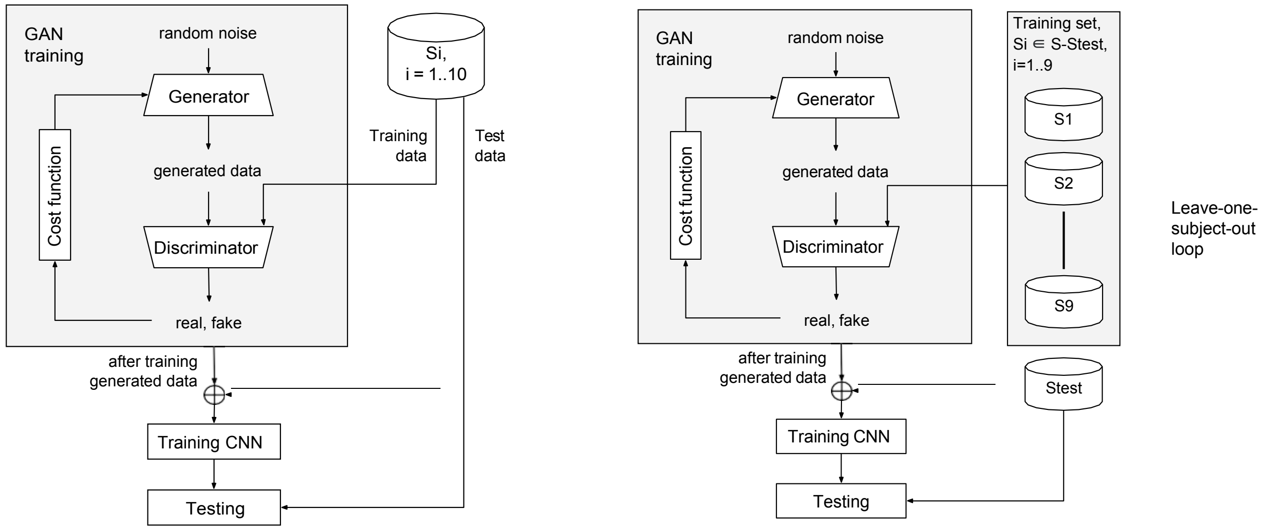
where $\mathrm{P}_{\hat{\boldsymbol{x}}}$ is an implicitly defined sampling distribution with samples $\hat{\boldsymbol{x}}$. In this article, we follow the recommendations in [29] and use the generative model architecture as in the DCGAN. We also disregard the batch normalization operators in the discriminator.

### C. Data generation methodology

In this study, we conduct three types of experiments to validate the generative models. First, we experiment to evaluate the quality of the generated data. We pool all subjects' data for each class and use it to train generative models and then generate equal numbers of samples. Second, we perform subject-specific augmentation, by using a certain amount of subject's data to generate samples. Third, we make subject-independent augmentation following the principle of leave-one-subject-out (LOO), in which we pool the data of all subjects but test subject to train generative models.

For all experiments, we draw random latent noise from Gaussian normal distribution to use as an input to the generator model. The data is normalized by applying a min-max normalization. Since the dataset has imbalanced class distribution, we only select and utilize the same number of non-target samples as target samples. For convenience during model design and training, we process only the first $76$ time points per channel instead of provided $78$. The reason is that we could build deeper generative models with $76$ points as it is a factor of 4. Therefore, the overall dataset size is $(2 \times 288 \times 10, 16, 76)$, where $288$ is the number of observations per class per subject, and $16$ is the number of channels per observation (see Section II-A).

We trained each generative model for both GAN models using a batch size of 32 for 500 epochs. For training DCGAN, we utilized Adam optimizer [30] with a learning rate of $10^{-4}$, and exponential decay rates of 0.5 and 0.999, as suggested in [24]. In addition, based on Eq. 1, the binary cross-entropy loss has been used. For training WGAN-GP, we also utilize Adam optimizer with a learning rate of $10^{-4}$ but with different decay




(a) Subject-specific

(b) Subject-independent

Fig. 1: Data augmented classification methodologies.

exponential rates of 0 and 0.9 [29]. We select $\lambda$ to be 10 and a critic (discriminator)-to-generator iterations to be 5.

### D. Evaluation of the generated data

*1) GAN-test:* For the first experiment with generative models, we perform the GAN-test used in [18]. We pool the entire data across subjects for each class and use it to train and further generate the same amount of artificial P300 data. We then train two classifiers on this real data and test them on the generated data. For classifiers, we use the linear discriminant analysis (LDA) [31] and deep learning-based convolutional neural network (CNN) [32].

*2) t-SNE visualization:* Using the real and generated samples from Section II-D-1, we utilize t-Distributed Stochastic Neighbor Embedding (t-SNE) to map the samples from a high-dimensional space to a lower one to visually assess and observe the data.

### E. Data augmented classification

We monitored the test classification accuracy for different amount of real data that were used in training stage. This practice helped us monitor the classification performance changes from a relatively small to large sample sizes. In this regard, $n$ samples were randomly drawn from the dataset available for each subject to train both the GAN and the final classifier (we use $n = 50, 288$). Furthermore, in order to monitor the effect of the size of artificially generated data, we perform classification for different ratio $k$ of the size of generated data to the real data ($k = 1, 2, 4$).

Hence, the training set size for (i) subject-specific classification is $(k+1) \times n$. in the case of the GAN-based augmentation and $n$ in the case of no augmentation; (ii) subject-independent classification is $9 \times (k+1) \times n$ in the case of the GAN-based augmentation and $9 \times n$ in the case of no augmentation.

*1) Subject-specific augmentation:* As the data for each subject includes $288$ observations per class, the half is used for training generative models. The generated samples are then attached to this training data to form an augmented set of size $(k + 1) \times n$, on which the classifier is trained. The second half of the subject's data is treated as an unseen test set (see Fig. 1a).

*2) Subject-independent augmentation:* By the leave-one-subject-out methodology, we hold one subject at a time as a test set and pool the rest of the subjects' data into one training set. The training set is entirely used to train generative models, hence to generate artificial P300s. The generated samples are concatenated with the training set of size $9 \times (k + 1) \times n$ and thus, the classifier is trained on the augmented set (see Fig. 1b).

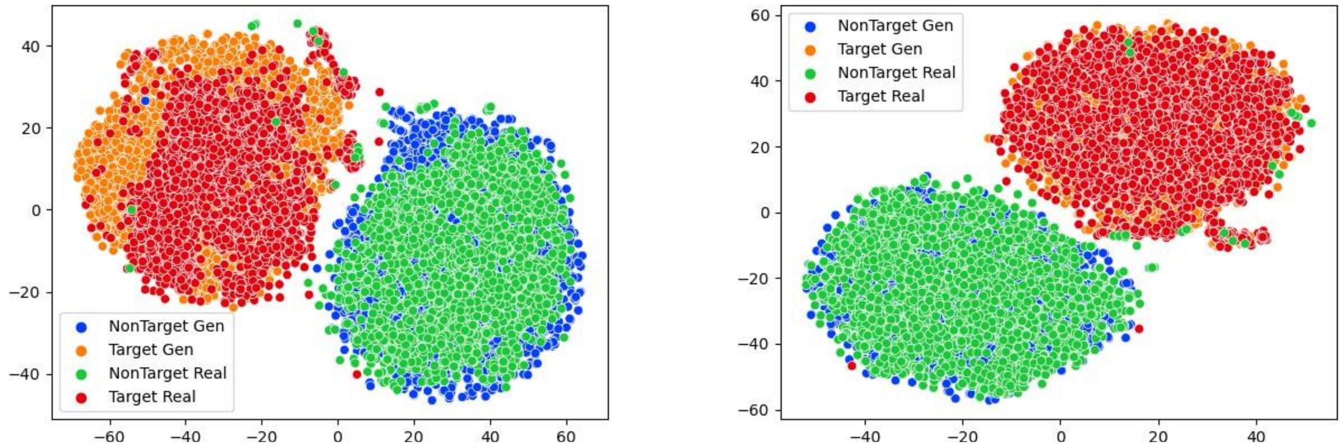
*3) Training the augmented data:* As for a classifier, the end-to-end convolutional neural network (CNN) [32] is exploited. The model is comprised of two 2D convolutional blocks, each followed by a 2D max-pooling layer and ReLU activation. After all the convolutional layers, the final feature map is applied dropout ($0.2$), flattened, and processed to a fully connected layer that has $2$ neurons at its output, followed by a sigmoid activation. The loss function and optimizer are Cross-entropy and Adam with a learning rate of $10^{-4}$. The training is performed on a batch size of $32$ for $300$ epochs to ensure convergence.

## III. Results

### A. Evaluation of the generated data

Table I reports the GAN-test accuracy (see Section II-D) for two generative models. The results show that both DCGAN and WGAN-GP models generated P300 data that could be classified with high accuracy using both tested classifiers.

[여기에 입력]




(a) DCGAN

(b) WGAN-GP

Fig. 2: t-SNE distribution plots for the real and generated data of each class.

TABLE I: GAN-test accuracies obtained using different classifiers. The classifiers were trained on the real samples and were tested on the generated samples of the same size.

| | LDA | CNN |
|---|---|---|
| DCGAN | 100% | 100% |
| WGAN-GP | 100% | 100% |

Similarly, Fig. 2 illustrates the t-SNE plots (see Section II-D) for the GAN-test data. As can be observed, the real and generated samples from the same class are grouped together, while different classes are clearly separated. The results observed in these plots corroborate the high classification accuracies achieved in Table I.

### B. *Data augmented classification*

For both subject-specific and subject-independent classification experiments conducted in this part (see Section II-E) and as a baseline for comparison, we also show the classification performance when no data augmentation has been used. Furthermore, to avoid possible biases in randomization, all experiments were repeated three times and the average accuracy was calculated and reported. Tables II and III show the subject-specific classification accuracy obtained on test data for each subject in experiments that were described in Section II-E. Table II shows that in general for a small amount of real data that was used in training stage (i.e., $n = 50$), the subject-specific classification performance achieved after data augmentation using both DCGAN and WGAN-GP and for different ratios of $k$ seems comparable to the case where no augmentation was used, although the results obtained using WGAN-GP augmentation seem slightly better than DCGAN. However, for the same scenario as we increase the size of the real data to $288$, both DCGAN and WGAN-GP data augmentation models lead to an evident improvement in terms of classification performance with respect to the case where no data augmentation was used (see Table III). In this scenario too, WGAN-GP shows a slightly better performance with respect to DCGAN. The observed margin of improvement reaches up to $11\%$ for some subjects, while on average ranges between $2\%$ to $4\%$. Furthermore, we observe that in this case, a higher $k$ would generally lead to a better classification accuracy.

Interestingly, the results of subject-independent classification tabulated in Tables IV and V portray a different picture for the effectiveness of DCGAN and WGAN-GP data augmentation models. In this case, the data augmentation models led to a better classification performance with respect to the baseline (i.e., no augmentation) when a relatively small size of the real data was used in training stage (i.e., $n = 50$). In contrast, for the relatively larger sample size (i.e., $n = 288$), none of the GAN-based models could outperform the baseline accuracy. Nonetheless, WGAN-GP-based augmentation still shows a better performance with respect to DCGAN.

## IV. Conclusion

In this work, we studied the effect of generating artificial P300 waves facilitated by generative adversarial networks (GAN). We utilized various qualitative tests such as GAN-tests and t-SNE visualization to validate the utility of GANs in synthesizing this type of data. Nevertheless, the ultimate goal of generating artificial data in classification of EEG signals is to improve the classification performance due to a limited amount of training data. Therefore, we examined the effect of generating artificial P300 waves on classification performance achieved in both the subject-specific and the subject-independent classification schemes.

[여기에 입력]

TABLE II: Subject-specific classification accuracy for generated to real data ratios $k = 1, 2, 4$ for a sample size of $n = 50$ where $n$ denotes the real data training size randomly selected from one training subject. Hence, the total training set size in the case of GAN augmentation is $(k + 1) \times n$ and in the case of no augmentation is $n$. Note that the average test accuracy denotes the average across repeating an experiment three times.

| | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** | **S7** | **S8** | **S9** | **S10** | **Average** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No augmentation | 67.71 | 71.53 | 69.44 | 65.28 | 70.49 | 58.33 | 65.62 | 53.82 | 72.57 | 72.22 | 66.70 ± 5.92 |
| | | | | | | **Ratio 1:1** | | | | | |
| DCGAN | 62.38 | 62.04 | 67.13 | 73.73 | 65.86 | 59.72 | 59.03 | 59.49 | 69.56 | 74.42 | 65.34 ± 5.47 |
| WGAN-GP | 65.51 | 64.7 | 69.1 | 69.1 | 70.14 | 56.94 | 62.5 | 61.81 | 71.76 | 73.38 | 66.49 ± 4.85 |
| | | | | | | **Ratio 2:1** | | | | | |
| DCGAN | 63.31 | 62.27 | 67.25 | 71.18 | 64.58 | 62.5 | 58.22 | 59.84 | 70.02 | 74.31 | 65.35 ± 4.93 |
| WGAN-GP | 65.28 | 68.52 | 68.29 | 66.9 | 73.96 | 61.0 | 63.54 | 60.19 | 71.18 | 74.31 | 67.31 ± 4.69 |
| | | | | | | **Ratio 4:1** | | | | | |
| DCGAN | 63.08 | 61.11 | 65.62 | 70.95 | 64.35 | 61.0 | 59.14 | 59.49 | 69.68 | 74.77 | 64.92 ± 5.03 |
| WGAN-GP | 62.04 | 65.97 | 66.09 | 68.52 | 73.73 | 61.69 | 60.88 | 62.73 | 72.8 | 75.58 | 67.00 ± 5.15 |

TABLE III: Subject-specific classification accuracy for generated to real data ratios $k = 1, 2, 4$ for a sample size of $n = 288$ where $n$ denotes the entire real data training size from one training subject. Hence, the total training set size in the case of GAN augmentation is $(k + 1) \times n$ and in the case of no augmentation is $n$. Note that the average test accuracy denotes the average across repeating an experiment three times.

| | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** | **S7** | **S8** | **S9** | **S10** | **Average** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No augmentation | 69.79 | 83.80 | 73.61 | 78.94 | 85.30 | 75.00 | 74.07 | 77.31 | 84.26 | 87.50 | 78.96 ± 5.65 |
| | | | | | | **Ratio 1:1** | | | | | |
| DCGAN | 74.88 | 82.06 | 75.93 | 80.09 | 86.92 | 78.70 | 77.66 | 79.28 | 87.62 | 88.31 | 81.15 ± 4.65 |
| WGAN-GP | 73.26 | 85.30 | 76.97 | 80.44 | 87.85 | 79.17 | 77.66 | 78.94 | 84.14 | 89.12 | 81.28 ± 4.85 |
| | | | | | | **Ratio 2:1** | | | | | |
| DCGAN | 76.04 | 81.71 | 76.27 | 80.79 | 86.57 | 77.20 | 76.97 | 77.31 | 87.50 | 89.00 | 80.94 ± 4.79 |
| WGAN-GP | 77.31 | 87.73 | 75.81 | 82.52 | 86.46 | 77.78 | 81.25 | 81.13 | 86.57 | 90.05 | 82.66 ± 4.62 |
| | | | | | | **Ratio 4:1** | | | | | |
| DCGAN | 77.55 | 83.91 | 75.12 | 82.41 | 85.53 | 77.66 | 76.85 | 79.40 | 89.12 | 89.58 | 81.71 ± 4.93 |
| WGAN-GP | 81.48 | 87.38 | 78.47 | 78.94 | 86.69 | 79.40 | 79.75 | 80.32 | 88.54 | 89.47 | 83.04 ± 4.19 |

TABLE IV: Subject-independent classification accuracy for generated to real data ratios $k = 1, 2, 4$ for a sample size of $n = 50$ where $n$ denotes the real data training size randomly selected from one training subject. Hence, the total training set size in the case of GAN augmentation is $9 \times (k + 1) \times n$ and in the case of no augmentation is $9 \times n$. Note that the average test accuracy denotes the average across repeating an experiment three times.

| | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** | **S7** | **S8** | **S9** | **S10** | **Average** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No augmentation | 66.49 | 63.19 | 68.4 | 76.39 | 76.39 | 68.58 | 69.1 | 69.97 | 83.51 | 77.78 | 71.98 ± 5.91 |
| | | | | | | **Ratio 1:1** | | | | | |
| DCGAN | 67.71 | 68.52 | 73.15 | 76.97 | 78.7 | 64.99 | 67.82 | 67.19 | 78.99 | 80.38 | 72.44 ± 5.55 |
| WGAN-GP | 66.49 | 75.35 | 71.3 | 75.29 | 76.1 | 68.98 | 66.96 | 69.27 | 80.73 | 79.17 | 72.96 ± 4.80 |
| | | | | | | **Ratio 2:1** | | | | | |
| DCGAN | 66.26 | 68.75 | 71.06 | 75.58 | 76.68 | 67.65 | 68.81 | 65.86 | 78.53 | 82.35 | 72.15 ± 5.44 |
| WGAN-GP | 67.07 | 75.35 | 70.89 | 75.17 | 74.42 | 68.52 | 67.01 | 69.73 | 79.92 | 78.94 | 72.70 ± 4.48 |
| | | | | | | **Ratio 4:1** | | | | | |
| DCGAN | 66.67 | 69.16 | 71.93 | 74.54 | 77.89 | 65.62 | 69.27 | 67.71 | 78.94 | 79.69 | 72.14 ± 5.01 |
| WGAN-GP | 67.42 | 75.06 | 71.41 | 73.9 | 74.65 | 68.29 | 66.84 | 70.25 | 79.63 | 78.18 | 72.56 ± 4.22 |

[여기에 입력]

TABLE V: Subject-independent classification accuracy for generated to real data ratios $k = 1, 2, 4$ for a sample size of $n = 288$ where $n$ denotes the real data training size randomly selected from one training subject. Hence, the total training set size in the case of GAN augmentation is $9 \times (k+1) \times n$ and in the case of no augmentation is $9 \times n$. Note that the average test accuracy denotes the average across repeating an experiment three times.

| | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | S10 | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No augmentation | 72.57 | 73.26 | 73.09 | 77.43 | 84.20 | 75.17 | 73.78 | 73.61 | 85.24 | 86.81 | 77.52 ± 5.36 |
| | | | | | | **Ratio 1:1** | | | | | |
| DCGAN | 70.66 | 75.98 | 73.44 | 78.99 | 82.64 | 73.38 | 69.73 | 70.49 | 85.19 | 85.13 | 76.56 ± 5.74 |
| WGAN-GP | 68.69 | 75.23 | 73.44 | 79.57 | 83.85 | 73.15 | 68.06 | 72.97 | 85.65 | 83.62 | 76.42 ± 6.04 |
| | | | | | | **Ratio 2:1** | | | | | |
| DCGAN | 68.81 | 72.45 | 74.02 | 79.17 | 82.41 | 71.99 | 70.25 | 69.44 | 82.87 | 83.28 | 75.47 ± 5.56 |
| WGAN-GP | 70.54 | 75.52 | 75.35 | 78.70 | 83.62 | 72.16 | 69.10 | 72.74 | 84.84 | 82.64 | 76.52 ± 5.37 |
| | | | | | | **Ratio 4:1** | | | | | |
| DCGAN | 67.88 | 73.21 | 73.09 | 78.24 | 81.66 | 70.43 | 68.75 | 69.50 | 84.49 | 83.56 | 75.08 ± 6.05 |
| WGAN-GP | 67.53 | 75.17 | 74.59 | 77.95 | 83.51 | 69.10 | 66.90 | 71.30 | 83.97 | 81.31 | 75.13 ± 6.09 |

We observe that a transition from a relatively small to large amount of real data (per subject) that were used in training our GANs and classifiers would generally lead to a better classification performance in subject-specific context; however, when a limited number of training subjects is available, this transition could potentially lead to an opposite effect in case of subject-independent classification. Nevertheless, rather than using these observations to draw a firm conclusion on the efficacy of GAN-based data augmentation for subject-independent classification, we would leave this as a hypothesis for further investigation in future studies. Nevertheless, to explain this rather counterintuitive observation in the subject-independent classification that was considered in this work, we note that a larger amount of real data per training subject would potentially lead to a generative model that is more specific to the latent space of training subjects. In this case, if the collected data for independent test subjects do not live in this learned latent space of representations, sampling from this space does not lead to a data with characteristics similar to those of test subjects. In contrast, a smaller amount of real data per training subject would potentially leave more room for latent space sampling performed by generative models, potentially covering the latent space of data from independent test subjects.

## REFERENCES


[1] B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K.-R. Müller, "Single-trial analysis and classification of ERP components—a tutorial," *NeuroImage*, vol. 56, no. 2, pp. 814–825, 2011.

[2] D. Nurseitov, A. Serekov, A. Shintemirov, and B. Abibullaev, "Design and evaluation of a P300-ERP based BCI system for real-time control of a mobile robot," in *Brain-Computer Interface (BCI), 2017 5th International Winter Conference on*. IEEE, 2017, pp. 115–120.

[3] A. Rezeika, M. Benda, P. Stawicki, F. Gembler, A. Saboor, and I. Volosyak, "Brain–computer interface spellers: A review," *Brain Sciences*, vol. 8, no. 4, p. 57, Mar. 2018. [Online]. Available: https://doi.org/10.3390/brainsci8040057

[4] B. Abibullaev and A. Zollanvari, "Learning discriminative spatiospectral features of ERPs for accurate brain–computer interfaces," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 5, pp. 2009–2020, Sep. 2019.

[5] S. Kundu and S. Ari, "P300 based character recognition using convolutional neural network and support vector machine," *Biomedical Signal Processing and Control*, vol. 55, p. 101645, Jan. 2020.

[6] L. Roijendijk, "Variability and nonstationarity in brain computer interfaces," Unpublished master's thesis, Radboud University Nijmegen, 2009.

[7] M. Krauledat, M. Tangermann, B. Blankertz, and K.-R. Müller, "Towards zero training for brain-computer interfacing," *PLoS ONE*, vol. 3, no. 8, p. e2967, Aug. 2008.

[8] F. Lotte, "Signal processing approaches to minimize or suppress calibration time in oscillatory activity-based brain–computer interfaces," *Proceedings of the IEEE*, vol. 103, no. 6, pp. 871–890, Jun. 2015.

[9] J. Fan, C. Sun, C. Chen, X. Jiang, X. Liu, X. Zhao, L. Meng, C. Dai, and W. Chen, "EEG data augmentation: Towards class imbalance problem in sleep staging tasks," *Journal of Neural Engineering*, Sep. 2020.

[10] M. M. Krell and S. K. Kim, "Rotational data augmentation for electroencephalographic data," in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, Jul. 2017.

[11] J.-H. Cho, J.-H. Jeong, and S.-W. Lee, "Decoding of grasp motions from EEG signals based on a novel data augmentation strategy," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, Jul. 2020.

[12] M. M. Krell, A. Seeland, and S. Kim, "Data augmentation for brain-computer interfaces: Analysis on event-related potentials data," *ArXiv*, vol. abs/1801.02730, 2018.

[13] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[15] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[16] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," 2016.

[17] Q. Zhang and Y. Liu, "Improving brain computer interface performance by data augmentation with conditional deep convolutional generative adversarial networks," 2018.

[18] F. Fahimi, S. Dosen, K. K. Ang, N. Mrachacz-Kersting, and C. Guan, "Generative adversarial networks-based data augmentation for brain-computer interface," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2020.

[19] N. Aznan, A. Atapour-Abarghouei, S. Bonner, J. Connolly, N. A. Moubayed, and T. Breckon, "Simulating brain signals: Creating synthetic eeg data via neural-based generative models for improved ssvep


[여기에 입력]

classification," *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2019.

[20] S. Panwar, P. Rad, T.-P. Jung, and Y. Huang, "Modeling EEG data distribution with a wasserstein generative adversarial network to predict RSVP events," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 8, pp. 1720–1730, Aug. 2020.

[21] P. Aricò, F. Aloise, F. Schettini, S. Salinari, D. Mattia, and F. Cincotti, "Influence of P300 latency jitter on event related potential-based brain–computer interface performance," *Journal of Neural Engineering*, vol. 11, no. 3, p. 035008, Jun. 2014.

[22] V. Jayaram and A. Barachant, "MOABB: trustworthy algorithm benchmarking for BCIs," *Journal of Neural Engineering*, vol. 15, no. 6, p. 066011, Sep. 2018.

[23] J. Guo, S. Lu, H. Cai, W. Zhang, Y. Yu, and J. Wang, "Long text generation via adversarial training with leaked information," 02 2018.

[24] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[25] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.

[26] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," 2017.

[27] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014.

[28] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," 2017.

[29] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5767–5777. [Online]. Available: http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[31] R. A. Fisher, *Statistical Methods for Research Workers*. Edinburgh: Oliver &Boyd, 1925.

[32] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[여기에 입력]